May 1988

A.E.Res. 88-7

THE ECONOMICS OF HATCHERY PRODUCED ALGAE AND BIVALVE SEED

by

Julia A. Myers

and

Richard N. Boisvert

Department of Agricultural Economics
Cornell University Agricultural Experiment Station
New York State College of Agriculture and Life Sciences
A Statutory College of the State University
Cornell University, Ithaca, New York 14853-7801

It is the policy of Cornell University actively to support equality of educational and employment opportunity. No person shall be denied admission to any educational program or activity or be denied employment on the basis of any legally prohibited discrimination involving, but not limited to, such factors as race, color, creed, religion, national or ethnic origin, sex, age or handicap. The University is committed to the maintenance of affirmative action programs which will assure the continuation of such equality of opportunity.

THE ECONOMICS OF HATCHERY PRODUCED ALGAE AND BIVALVE SEED

by

Julia A. Myers and Richard N. Boisvert

ABSTRACT

The bivalve aquacultural industry in the northeastern United States has been strengthened by declining and fluctuating natural harvests in the surrounding coastal region. Hatcheries in which the bivalves are grown to a small seed size have gained acceptance as a link in the aquacultural process. These firms are potentially able to provide a consistent, reliable source of seed to the bivalve aquaculturist.

This research contributes to an understanding of the hatchery industry by examining factors that influence production costs. The research is based on observations of a working hatchery located in the northeast and on current literature. The data were used to develop the framework for a computer program that can estimate variable production costs. This program required the development of an algorithm that simulates the production process. The model sums costs at different stages of animal growth, which are further broken down into smaller units of time.

There was sufficient information available from the literature and hatchery observations to simulate the overall production system, but expenditure records did not separate costs between stages or between hatchery and open-water field operations. The empirical analysis focuses on estimating costs of algae production, because data on this process were more readily isolated from the rest of the hatchery and because feeding costs are recognized as a large proportion of total costs.

The FORTRAN computer program based on the model focuses on the cost of producing algae feed, but as more details become available about other hatchery production processes it would be a simple matter to incorporate them into the existing FORTRAN code. The computer program combines estimates of the cost of algae production with survival and growth rates and feeding efficiencies to determine the costs of feeding a batch of bivalves to various sizes.

The model is applied to raising Atlantic Oyster seed. Twelve initial simulations were made, assuming different values for important parameters. To explore the implications of these feed cost simulations for the marketing of bivalve seed, some rough estimates of total costs of production per bivalve were developed on the basis of others' estimates of the proportion that feed costs are of total costs. Under the most optimistic assumptions, bivalve seed can be produced and marketed profitably. These conclusions must be qualified by the fact that sufficient data were not available from the hatchery under study to estimate all components of total cost. Collection of these types of data would help in further testing of the algorithm as well as contributing to an overall understanding of hatchery costs at each stage of production. Equally important from a research perspective is the need for better data on hatchery survival rates and the size-age and feeding rates.

Table of Contents

List of Tables

List of Figures

THE ECONOMICS OF HATCHERY PRODUCED ALGAE AND BIVALVE SEED

by

Julia A. Myers and Richard N. Boisvert*

*I - INTRODUCTION**

Over the past 25 years, bivalve aquaculture in the northeastern United States has become a reliable supplier of marketable oysters and clams. However, to initiate production, the industry requires adequate supplies of bivalve seed. Seed can be supplied from either natural sets or artificial spawning in a hatchery.

A hatchery operator's ability to produce and market seed as an efficient alternative to naturally-spawned seed requires accurate information about the market, the production process and production costs. As with most managers in a new, rapidly developing industry, hatchery operators have difficulty assembling adequate information to make effective investment and production decisions.

The research reported here contributes to an understanding of the economics of artificially spawned bivalve seed by: a) describing the hatchery industry's development and its production processes and b) formulating a model, associated algorithms and computer code by which operators may calculate variable production costs.

Much of the information about production processes, input requirements and prices was obtained in 1983 from one hatchery on the northeast coast. That being its first full year of operation, the firm's record keeping system was new and not well enough developed to facilitate accurate measurement of the quantities of inputs used at every stage of a bivalve seed's development. However, feed contributes importantly to the total variable cost of bivalve seed (Im *et al*. 1976; Gates *et al*. 1974 and Bolton 1982). Data on algae production were quite good. Thus, the algorithms that calculate algae production and feeding cost are developed in detail. The algorithms can also be used to examine the sensitivity of results with respect to different assumptions about survival rates, growth rates and feeding efficiency.

The remainder of the bulletin begins, in Section *II*, by placing the hatchery industry within the context of the historical decline in the natural bivalve harvest. Various aspects of the hatchery industry affecting the firm and its costs, such as the industry's information network, the issue of property rights and market structure are also discussed. Section *III* describes the production processes using one hatchery as an illustration. This section also outlines choices that affect quality, quantity and timing of inputs and outputs and contains a general discussion of production costs. Section *IV* describes algorithms that simulate the cost of production for each of the system's components. Section *V* presents the empirical application of the program to the American Atlantic Oyster, including tests using data obtained from the hatchery and sensitivity analyses for the program's important variables. Preliminary estimates of total production costs per bivalve are compared with bivalve seed prices to determine at what age (size) the bivalve seed should be sold. Section *VI* summarizes the major findings and their implications for a hatchery's production.

---

*J. A. Myers is a Research Associate at the University of California, San Diego, Laboratory for Mathematics and Statistics, and R. N. Boisvert is a professor in the Department of Agricultural Economics, Cornell University. The research draws extensively on Myers' thesis (1985) sponsored by the New York Sea Grant Institute under a grant from the Office of Sea Grant, NOAA. The cooperation from individuals at the hatchery where data were collected is gratefully acknowledged. The opinions expressed in the report are those of the authors and not necessarily those of the funding agency nor other cooperators in the research.

II - FORMATION OF THE HATCHERY INDUSTRY

In the northeastern United States, the long-standing clam and oyster fishing industry has given rise to the development of the bivalve hatchery industry. Over harvests and increased urbanization, with the consequent decline in the once-abundant natural bivalve population, have forced harvesters to look for alternatives. Among these alternatives are government regulation and incentives, but perhaps the most important alternative is the aquacultural industry, which is replacing traditional methods of harvest. Recently, hatcheries producing bivalves to seed size only have emerged as an alternative to natural bivalve spawnings and also as a link in the aquacultural production of marketable bivalves.

Background

Clams and oysters are harvested in all 14 states along the Atlantic Seaboard (U.S. Dept. of Commerce 1979). The most common oyster is the American Atlantic Oyster (*Crassostrea virginica*). Four species of clam--hard, soft, surf, and ocean quahog--constitute 99% of total U.S. landings by weight. The northern hard shell clam (*Mercenaria mercenaria*) accounts for 53% of the landings, although it constitutes only 17% of their value (Dressel and Fitzgibbon 1978).

Long Island, New England, and the Chesapeake Bay have historically been centers of clam and oyster fisheries (Ritchie 1977; Manzi *et al.* 1982b). Long Island Sound (the waters between Long Island and Connecticut), and Great South Bay, (the waters between Long Island and Fire Island), are the most productive of the New York areas (Figure 1). The seabeds there are hard, shallow and scattered with culch material to which the young animals attach (Korringa 1976). Additionally, the strong tidal currents carry abundant food into the bay (Bardach *et al.* 1972) and the bivalves are protected from many predators because the salinity is lower than in the surrounding ocean water.

Bivalves, especially oysters, have been an important food source for the New York Bight coastal region. Oysters were used by Native Americans for food and trade and later by the colonialists (Terry 1977). Clams were fewer in number than oysters, but by the 1900's the rise of the cannery industry encouraged the growth of the clamming industry (Dressel and Fitzgibbon 1978).

The rise in clam production was interrupted after red meat shortages of the 1940's put increased pressure on the shellfish industry. Landings in the New York Marine District decreased from 6.8 million to 3.3 million pounds between 1943 and 1945 (Figure 2). Although landings later increased, the industry continued a pattern of boom and bust. In 1976 Long Island harvests reached a peak of 8.4 million pounds, worth about $16.8 million (Freedman and Morris 1983). Great South Bay was considered the most important hard-shell clam industry in the world. It harvested 80% of Long Island's total landings, which represented 50% of the U.S. total (U.S. Dept. of Commerce 1974). Since then, the Long Island clam harvest has declined; by 1982, harvests were down by 60% from 1976 levels (Figure 3).

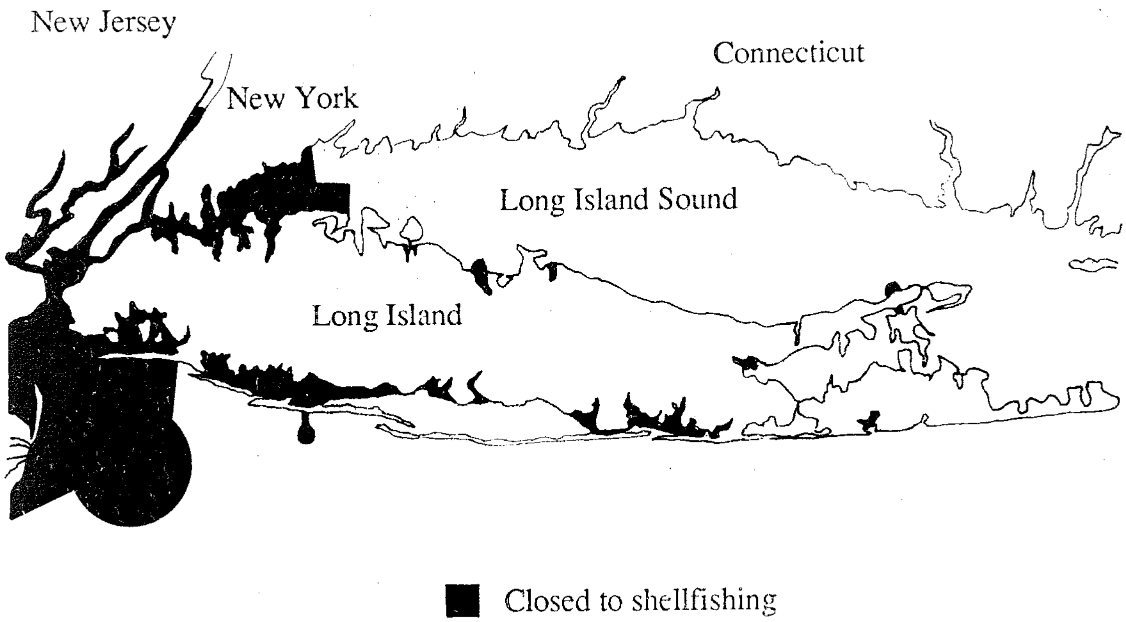
After a peak in 1904, oyster harvests in New York remained fairly stable between 1921 and 1952 but experienced a near-collapse in the 1960's (Figure 2). Revival of the industry through aquaculture has increased landings. Today, oyster production in the New York region is based entirely on aquaculture (Terry 1977).

Population Dynamics

To understand the dramatic changes in the Long Island clam and oyster industries, one must examine the interaction among competing factors that determine the cyclic pattern of bivalve populations. Although population fluctuations occur naturally, they have been exaggerated through over harvest and destruction of natural habitat.

An inherent problem of common property fisheries is overuse (Hardin 1968), which leaves too few animals unharvested for the species to repopulate. This problem is acute around Long Island because bivalves in

FIGURE 1. MAP OF LONG ISLAND COASTAL REGION



Source: Terry (1977)

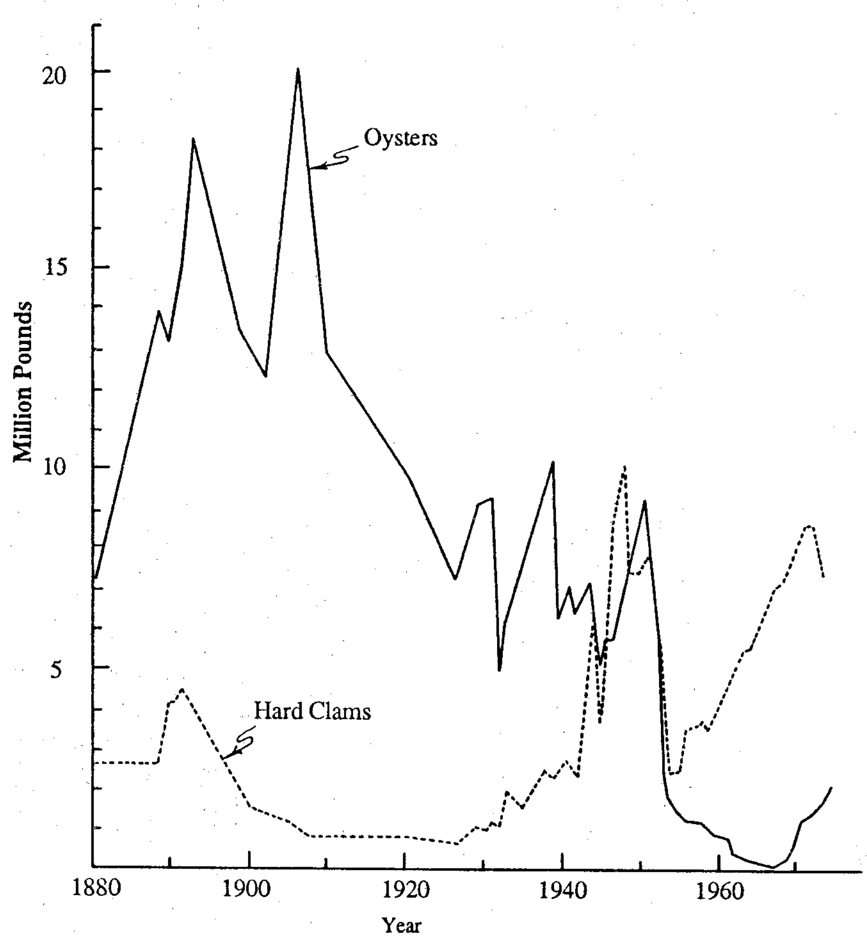
the cool waters require a long maturation period and because a large proportion of the population must be left unharvested in order to sustain the population. Thus, one major reason for the periodic decline of the clam and oyster populations is the likelihood that harvests will occasionally exceed the maximum sustainable yield.

Competition with man has decreased the natural habitat of clams and oysters and adversely affected bivalve populations. As urbanization and shoreline use in the New York area intensified, natural spawning grounds were reduced. Siltation, which is particularly harmful to young spat, increased; and saltwater intrusion caused by the removal of sandbars decreased protection against natural predators (Freedman and Morris 1983). Because toxins accumulate in the flesh of oysters and clams, the increased sewage effluent and agricultural runoff in the area have increased mortality rates as well as decreased the marketability of the bivalve. The actual numbers of oysters and clams available for harvest have been reduced further through the closure of contaminated natural spawning beds (Figure 1).

Demand Considerations

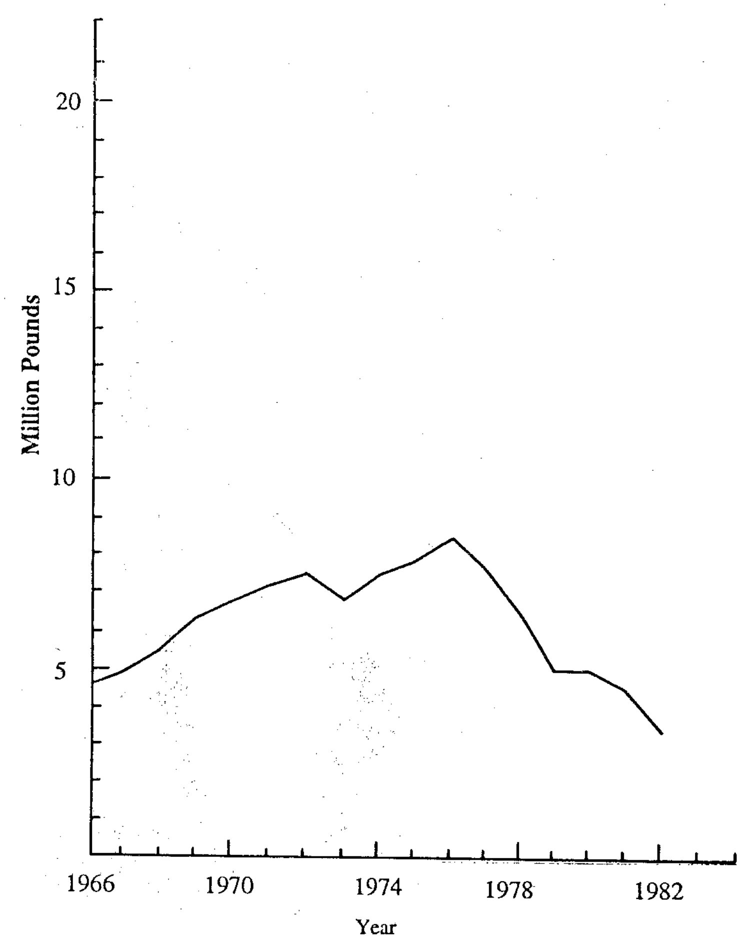
While many factors affecting the bivalve population dynamics have resulted in a reduction in their supply, the industry has also been affected by demand considerations. For example, although the income elasticity of demand for clams is positive, it is relatively low, giving rise to only a slow growth in demand for clams over time due to rising per-capita income and increasing human population (Gates *et al.* 1974). The income elasticity of demand for oysters is positive as well, but per capita consumption has actually decreased

FIGURE 2. COMMERCIAL LANDINGS FROM NEW YORK MARINE DISTRICT WATERS



Sources: McHugh and Williams (1976); Terry (1977); Regional Marine Resources Council (1974)

FIGURE 3. LONG ISLAND CLAM HARVEST



Source: Freedman and Morris (1983)

(Im and Langmo 1977). This decrease might have been caused by either the decrease in supply (Gates *et al.* 1974) and the associated rise in price, or by a change in consumers' tastes due to the fear of eating contaminated animals.

Alternatives

These problems have led to a disappearance of, or change in, the traditional shellfish industry. The scarcity of the hard clam has caused a shift to other species, such as surf clams (Dressel and Fitzgibbon 1978). Production has shifted southward to less-populated regions where competition for habitat is not so keen (Manzi *et al.* 1982a). Also, oyster imports have increased (Im and Langmo 1977).

Options to halt the gradual disappearance of the existing bivalve industry do exist. One option is a comprehensive government management plan to stop the decrease in, or even increase, the natural bivalve populations. Aquacultural production of marketable bivalves and hatchery production of bivalve seed are also viable alternatives, but they both require radical change in traditional harvesting techniques.

Regulation

Because most clam and oyster beds are common-property resources, they have suffered, as most common-property resources have, from over-use and abuse. Although size, time and method of catch are regulated, no consistent policy for predator control or return of culch material to the beds has been in force (Matthiessen 1970). Furthermore, these regulations have often been ineffective because of inadequate knowledge, failure to enforce existing laws, lack of cooperation and an absence of comprehensive planning.

The Regional Marine Resources Council's Shellfish Committee, created in the 1970's, partially filled the void by identifying 18 guidelines for improving the industry (Regional Marine Resources Council 1974). A Long Island clam union exists to aid members of the industry. On the consumer side, identification of potentially harmful pathogens, public health inspections and enforcement of closed beds will help maintain consumer confidence and increase demand.

To protect the industry, some Long Island governments have purchased young shellfish seed in order to seed public grounds (Smith 1982-83; Goldstein 1983). The population could also be increased by controlling predators, working the bottom for better spawning success, transferring animals to non-polluted grounds and controlling pollution, saltwater intrusion and siltation of habitat.

Unfortunately, implementation of such non-traditional resource policies is politically arduous and requires placation of special-interest groups and cooperation among affected parties. These measures are often expensive to establish and to enforce. The overall benefits to society from an increased bivalve population and the resulting strengthened harvesting industry must be measured against the costs of implementing any resource-regulating mechanisms.

Aquaculture

Aquaculture, the growing of organisms in water under controlled conditions, offers an alternative to traditional bivalve harvesting techniques. The primary benefits of controlling the environment are increased product supplies, although factors adversely affecting quality, such as predators, crowding and parasites, may be reduced.

The viability of an aquaculture industry may depend on the granting of exclusive rights to harvest or to develop a section of the ocean bottom aquaculturally. Marine resources have traditionally been common property (Terry 1977), but sedentary marine life on the bottom may be considered private property, although this right is difficult to substantiate (Hanson 1974; Terry 1977).

The states around the New York Bight area and towns on Long Island have a history of granting bottom leases to private individuals (Kochiss 1974; Terry 1977; Gates *et al.* 1974). In New York, the Department of Environmental Conservation grants leases on the condition that the water quality is good and that no naturally-productive shellfish beds exist (Terry 1977). Great South Bay is held in trust by local townships; harvesters have full bottom rights to certain areas.

In general, economists support private ownership to increase efficiency and productivity (Agnello and Donnelley 1976; Terry 1977). Private beds experience five to ten times greater productivity (Dressel and Fitzgibbon 1978), probably because private harvesters can use less labor-intensive methods than can the public harvesters, who are required by the government to use inefficient harvesting techniques to control harvest quantity. Furthermore, without the assurance of exclusive rights to a harvest, there is no incentive for private harvesters to return culch, to control predators, to establish seed beds or to purchase seed to supplement the natural sets.

As stated above, the viability of the industry is demonstrated by the fact that all oysters on Long Island are currently produced by aquaculture (Terry 1977; Pillay 1976), and all oyster aquaculturists produce some hard clams (Korringa 1976). However, there has been increasing opposition led by local harvesters to the leasing of marine bottom, even of non-productive beds. Few leases have been granted recently (Terry 1977), and if these beds continue to remain in public control and open to all harvesters, the community must either exert better management controls or expect diminishing supply. On the other hand, private ownership might lead to an increase of marketable species but a decrease or disappearance of other naturally-occurring species.

Hatcheries

Potential advantages to hatchery-produced seed include greater reliability of production and better quality control. The development of the hatchery industry (which grows clam and other larvae into bottom-dwelling seed large enough to be planted in open waters for later cultivation or harvesting) has come in response to natural reproduction failure in open water (Loosanoff and Davis 1963a; Henderson 1978). Hatcheries can potentially achieve better quality control, decreased mortality and faster growth (Bardach *et al.* 1972; Donohue *et al.* 1981), thus offering aquaculturists a reliable source of bivalve seed. Furthermore, local governments can supplement declining natural spawns with hatchery-produced seed, a practice local harvesters generally favor as an alternative to direct governmental support of open-water growout in areas where a few individuals can affect the local bivalve market (Miller 1977; Rhodes 1974).

Worldwide demand for oyster and clam seed is growing at an increasing rate because of decreasing natural stocks and improving hatchery techniques. Henderson (1978) has estimated the annual world market for hatchery seed to be $2 billion, or 25% of the annual U.S. whole clam and oyster harvests. In the U.S., hatcheries exist in the Southeast, the Northwest and in parts of the Northeast. The U.S. leads in seed production, having pioneered many clam hatchery techniques, but its oyster techniques are still considered "primitive" (Bardach *et al.* 1972).

Opinions differ as to the sufficiency of seed availability. Clam seed is more readily available than oyster seed (Terry 1977), except for seed larger than 10 mm (Castagna and Kraeuter 1977; Manzi *et al.* 1982a). Supplies of oyster seed have proven unreliable and insufficient to meet demand at current price (Donohue *et al.* 1981; Im and Langmo 1977). The demand for hatchery-produced seed is cyclical and increases during natural set failures (Henderson 1978). If hatcheries could produce a reliable, inexpensive product that is more attractive to growers than natural seed, they could capture a larger share of the seed market.

Hatcheries currently operate with less than complete information for two reasons: a) lack of cooperation between affected parties in the industry (Henderson 1978); and b) inasmuch as no two hatcheries are operated exactly alike (Terry 1977), such information as is available from one is not entirely applicable to another. Partly due to this lack of information, hatcheries are only marginally cost-effective (Terry 1977). Cooperation should be improved between hatchery operators and buyers, researchers and the government (Henderson 1978). This in turn should affect the flow of information and help industry development.

Despite the problems faced by hatcheries, many in the field believe that they will continue as a viable industry (Terry 1977). Genetic breeding and development of domestic bivalve strains are future potentials of the industry (Terry 1977). The development and success of the hatchery industry depends on the continued demand for seed, encouragement by government and the scientific community and the success of hatchery operators. The success of the individual operator depends in large part on an ability to compete in the market place. To be competitive, the hatchery operator must be able to make the best production decisions based on an accurate assessment of production costs. This is the subject of the following sections.

III - HATCHERY PRODUCTION OF BIVALVE SEED

Raising oyster and clam seed under controlled conditions requires technically-sophisticated, delicately-monitored production processes. The purpose of this section is to describe these processes, using the hatchery from which data were collected as a model. Although production systems vary among hatcheries, especially in algae and culch preparation and in facility and system designs, most operations are universal to all hatcheries. Techniques are varied primarily to take advantage of available natural resources at the site.

The Hatchery

The bivalve seed hatchery, from which data were collected for this study, was the outgrowth of an experimental plant begun in the late 1970's. This model facility was used to help train staff and develop techniques utilizing the available natural resources in the surrounding area.

Construction of the permanent hatchery was completed in 1982. The building used for seed production has approximately 500 square meters of indoor floor space. There is an ample outdoor work area, as well as room available for expansion. The hatchery began partial production in 1982, and the first full production year was 1983. Both oysters and clams are grown, but production is now concentrated on oysters.

The hatchery is located a few hundred yards from a bay. The ownership of the bottom rights to part of the bay led to the development of a field operation ("outgrow") where some of the seed produced in the hatchery is grown to maturity. Most of these outgrow operations are conducted separately from those of the hatchery.

The Hatchery's Production System

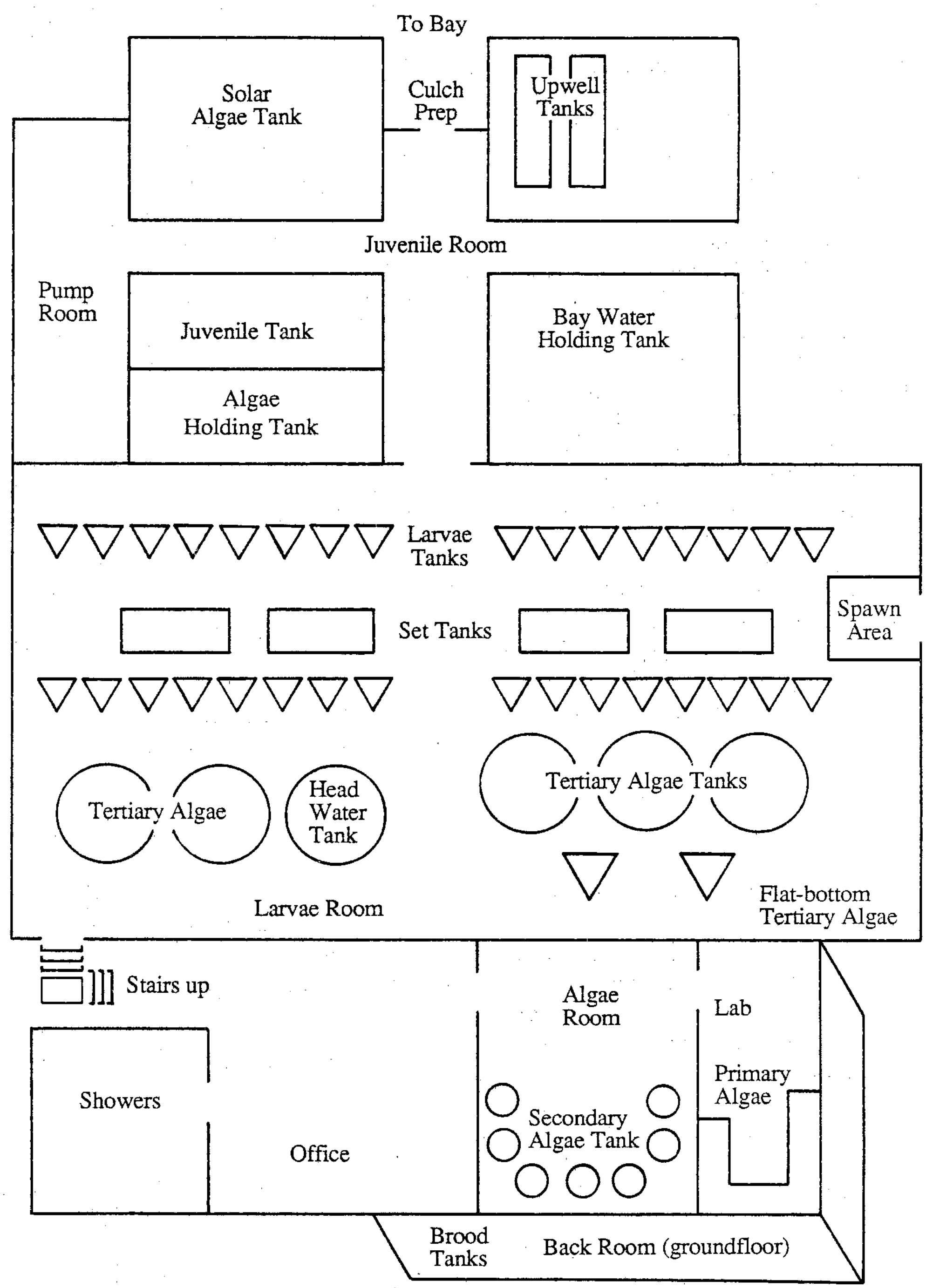
The production process can be divided into six life-cycle stages for the bivalve:[1] 1) broodstock maintenance and conditioning; 2) spawning; 3) larvae development; 4) setting; 5) juvenile development and 6) field (or outgrow); and two support stages: 1) algae production and 2) culch production. Figure 4 shows the location of each production stage within the hatchery.

To maintain a low mortality rate, young bivalves must be treated with care during all phases of production, including inspection, cleaning and transfer. Exposure to severe environmental conditions or fouling by foreign organisms, such as fungus, bacteria and predators, can cause death or deformation of the animals or a poisoned product. Thus, particular attention must be given to cleanliness at each stage of production and to selec-

---

[1] Because the differences between clam and oyster production are minimal, the two processes will be treated as one throughout the remainder of this analysis.

FIGURE 4. PRODUCTION LOCATIONS IN THE HATCHERY



tion of non-toxic materials for all equipment that comes in direct contact with the animals. The quantity and quality of inputs used, such as energy, algae, culch and other materials, depend on the biological requirements of the bivalve, the existing environmental conditions and on the hatchery's ability to purchase or produce those inputs. Inadequate or poor quality inputs will affect the survival of the bivalves.

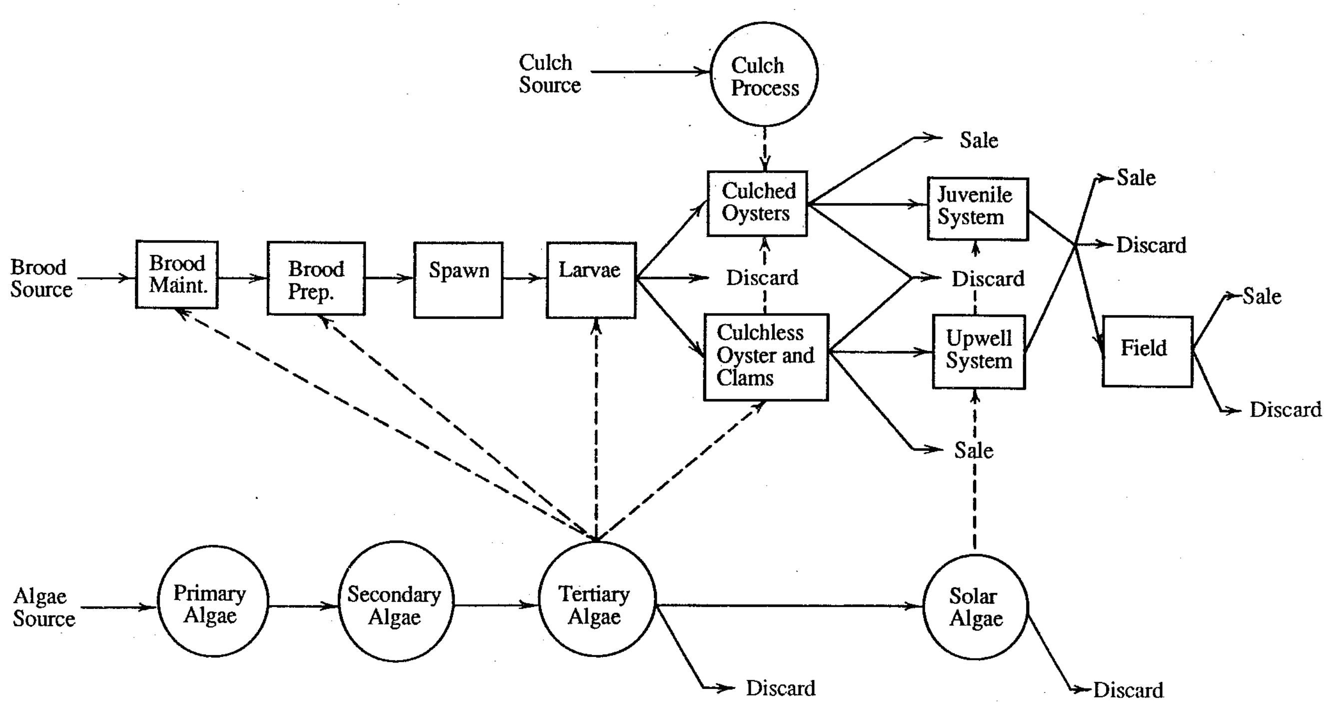
As with most biological production processes, there is a trade-off between achieving maximum production and the cost of inputs. For instance, higher densities improve efficiency in the use of inputs such as water and labor; however, a high density decreases the survival and growth rates of the existing animals. The hatchery must weigh the marginal cost of providing ideal environmental conditions against the resulting increase in the value of the marginal output.

Proper timing of production is critical in two respects. First, if the hatchery begins production too early in the season, the animals will be ready for transfer to the juvenile tanks or outgrow facility before the water has reached a suitable temperature. Extending production too late may leave the hatchery with unsold inventory. Second, new spawnings and transfers between stages must be timed so as not to exceed the system's capacity in terms of space or ability to produce sufficient inputs. The optimal age of transfer depends in part on the cost of maintaining the animal. The best way to understand the production system is to examine each of the components separately. (Figure 5 shows the flow between the production stages.)

Algae

Production of algae for feed in sufficient quantity and quality is important to the bivalve at all stages of growth and is one of the principal problems in profitably growing animals to market size entirely within a closed system (Gates *et al.* 1974; Epifanio 1975). Not much is known about the specific food requirements of the bivalve (Galtsoff 1964), especially about the adult bivalve (Epifanio 1975), but in the work that has been done, bivalve algae consumption is measured either by the number of cells consumed per animal or by the algae concentration made available to the animal. Existing estimates of the number of cells consumed by

FIGURE 5. FLOW BETWEEN HATCHERY PRODUCTION STAGES



oysters and clams at different stages of animal growth are in Table 1. Using the second and most common method, optimum algae concentrations for oyster larvae are given in Table 2.

The amount of algae assimilated, i.e., actually used for growth by the animal, is estimated at between 70% and 90% of the amount consumed (Epifanio 1975). Consumption is less than the filtering rate (the cells filtered per volume of water pumped), and the filtering rate in turn is less than the pumping rate. Filtering efficiency is affected by environmental conditions and was measured at between 34% and 58% by Baab and Associates (1973).

For bivalve feed, a mix of algae species is preferable (Ukeles 1975). The digestibility and nutrient value of the food depends on the chemical composition, size, texture, taste (Epifanio 1976), cell wall and toxicity (Ukeles 1975) of the algae species. Juveniles are less sensitive to algae quality and size than are larvae, which require cells sized from 6-10 microns in diameter (Breese and Malouf 1975). Clams are generally less particular than are oysters about the quality of algae fed (Ukeles 1975).

The quantity of algae that the hatchery must produce to feed the bivalves depends on whether or not sea-water containing naturally occurring algae is made available to the animals. The proportion of its diet an animal is able to receive in this way depends on the concentration of algae in the seawater, on the pumping and filtering abilities of the animal, on the seasonal variation of that concentration and on the amount of water to which the animal is exposed. Also, some additional algae must be produced and held in reserve in case main supplies are contaminated. Producing excessive amounts can be costly, because algae can be stored only a few days.

The hatchery's algae stage is managed by a single employee, who spends more than one-half the total working hours exclusively

Table 1. Daily Consumption of Algae Cells by Oysters and Clams

Stage/Age of Oyster	Equivalent Stage at the Hatchery	Algae Cells Consumed Per Day: Oysters	Clams[a]
Larvae			
2 days		1.0×10^4	
12-15 days	Larval	4.0×10^4	
set size		5.0×10^4	
Spat		2.0×10^5	1.0×10^5
3 weeks	Setting	3.0×10^6	
8 weeks		2.4×10^7	
Seed	Juvenile	8.0×10^7	6.0×10^5
Adult	Juvenile/Outgrow		1.0×10^7
3 inches		1.1×10^8	

Sources: Claus and Adler (1970); Matthiessen and Toner (1966); Gates et al. (1974).

[a]The three consumption levels for clams refer to animals of 0.2, 2.0 and 8.0 mm in size, respectively. The corresponding stages for the oysters listed at the left are probably not exactly comparable to these sizes.

Table 2. Optimum Algae Concentrations for Oyster Larvae

Initial Larvae Shell Length (μ)	Food Concentration for Highest Growth Rate (μl of packed cells/1)
77	2.5
78	2.5
104	10.0
104	10.0
139	10.0
146	20.0
201	40.0

Source: Rhodes and Landers (1973).

on algae production. The batch method, in which algae is cultured to a maximum density using successively larger containers, is used. Not all algae at a maximum age and density are used exclusively for food; instead, some are used to innoculate a larger tank.[2] This "modified batch" strategy requires additional labor (Matthiessen and Smith 1979).

The second-floor laboratory is where algae samples are analyzed for contamination and proper densities, nutrient solution is prepared to supplement the algae cultures, and initial algae stock is stored. Four different varieties are cultured: *Thalassiosira pseudoma* (diatom), *Pavlova lutherii*, *Pavlova lutherii Hoptaphyceae* and *Isochrysis galbana*. The latter represents 50% of the algae grown and 50% of the animal's diet. Because young animals require small algae cells, the diatom is used only during the juvenile stage.

The primary algae production stage begins with tube cultures. Algae, in nutrient-rich well water, are allowed to mature two weeks to a maximum density of 3 x 10^5 cells/ml. The culture is then transferred to sterilized flask containers. After one week, this culture matures and is transferred to carboys. Water must remain cool and sufficiently aerated. Cool fluorescent lighting provides the algae with artificial sunlight 24 hours each day.

The secondary stage, located in the algae room next to the laboratory, consists of seven 757-litre tanks. Each tank is innoculated with 16 litres of primary algae culture; nutrients and well water are added. The number of tanks used depends on near-term food requirements. Optimum algae density of 4 x 10^5 cells/ml is reached in one week.

The secondary culture flows to the ground floor to innoculate the 3,785-litre tertiary algae tanks. The requirements at this stage are similar to those for previous cultures; and when a maximum density of 5 to 8 x 10^5 cells/ml is reached, the tertiary, or "direct-feed", stage is used to feed the bivalve broodstock and the animals at the larvae and set stages. It is also used to innoculate the next culture stage. After one week, any algae remaining are discarded.

The final algae tank is a sunken 49,205-litre tank in the juvenile room. This "solar" tank is innoculated from the tertiary tanks and allowed to bloom under natural sunlight provided by large, sloping windows over the tank. Maintenance is similar to that in other stages. When a suitable density is reached, the entire amount is transferred to the algae holding tank located in the same room; the solar tank is then reinnoculated. The juvenile bivalves are fed directly from the holding tank. After one week, the remaining algae are discarded.

Broodstock Maintenance and Conditioning

The quality of the bivalve produced depends to a degree on the genetic quality of the parental stock, which is selected on the

[2] In contrast to techniques that centrifuge large quantities of seawater to concentrate naturally-occurring algae, this method selects only the best species of algae. The centrifuge process, which is commonly used on Long Island, is less expensive (Matthiessen and Smith 1979), but it requires an abundant and reliable source of naturally-occurring, desirable algae. The batch or modified batch process offers certain advantages for feeding efficiency because higher densities of algae can be obtained. With these processes, algae can be produced at densities of 10^7 cells/ml, as compared to densities in seawater of 10^4 cells/ml (Gates et al. 1974).

basis of size, shape and growth potential. The broodstock is kept in rectangular 825-litre tanks of aerated, filtered water. Cool water temperatures are maintained to prevent spontaneous reproduction.

Before spawning can occur, the mature adults must be "conditioned"; the water temperature is increased to 26-29°C, and the food supply is also increased. The conditioning time varies according to the temperature and the amount of food used, but oysters require approximately 35 days (Gates *et al.* 1974).

Spawning

Once the male and female bivalves have been conditioned, the spawning process takes two employees one to three hours to complete. Conditioned adults are induced to spawn by either introducing a sharp temperature increase or by adding previously-collected sperm (Breese and Malouf 1975). One female clam can produce as many as 24 million eggs at a time (Loosanoff and Davis 1963b). Within 24 to 48 hours, the gametes become free-swimming larvae.

Larvae Development

During the larvae stage, algae-rich seawater is circulated about the mobile animals ("veligers"). In the larvae room are 38 conical, 340-litre larvae tanks and two 700-litre, flat-bottom larvae tanks. These tanks are filled with warm, filtered seawater, which must be heated when seawater temperature is below 25°C. Approximately 14 litres of algae are added to each tank/day.

Oyster and clam larvae are maintained close to ideal densities for fastest growth and survival, viz., densities from 0.4 to 15 animals/ml and 5 animals/ml of water, respectively (Gates *et al.* 1974; Matthiessen and Toner 1966). Growth and survival rates at this and subsequent stages are affected not only by density and by food quantity and quality, but also by the temperature and available oxygen. Thus, water is maintained at the ideal temperature, and aeration is supplied continually to all tanks.

Maintenance of the larvae is labor-intensive. Besides daily feeding and monitoring, every 48 hours the tanks are cleaned, the larvae are sorted by size, and the dead animals are removed. This takes two employees nearly a full day to complete, depending on the number of tanks in use. Animals of similar size are collected together from different tanks and are then returned to a newly-filled tank, and algae are added. This culling process is an essential feature of the production process because it, in large part, determines the quality and the quantity of the final output.

The most critical period for the larvae occurs at metamorphosis, when the bivalves change from free-swimming larvae to bottom-dwelling spat. Setting occurs at 7 to 10 days for clams (Bardach *et al.* 1972; Gates *et al.* 1974) and 10 to 20 days for oysters (Gates *et al.* 1974). The clams at this stage measure 200 to 215 microns (Bardach *et al.* 1972; Gates *et al.* 1974) and oysters 300 to 325 microns. Prior to this metamorphosis, the animals must be transferred to the set tanks, which have been filled with culch.

Culch

Culch is the material onto which the oyster larvae set and attach permanently. Thus, before the larvae can be transferred to the setting tanks, culch is prepared. The choice of culch material depends on the type of grow-out anticipated rather than the relative cost of the materials (Matthiessen 1970; Gates *et al.* 1974).[3] Before 1960, whole oyster or clam shell was used exclusively for bottom culch material (Matthiessen and Smith 1979); other techniques, such as the suspended-string method, could be used in lieu of bottom culch. Today, choices for culch include a variety of shell sizes, artificial materials and "culchless" oysters.

The culchless oyster is an oyster with no attached culch, that is, one that has been induced to set on a material, such as plastic

[3] Clams do not require culch since they do not attach firmly.

sheeting, from which it can be removed easily. Culchless oysters can be grown at greater densities than the culched variety inasmuch as clumping, which crowds and deforms the spat on the culch, is less of a problem. Culchless oysters may be less expensive to produce because culch production costs are avoided, and because shipping and handling of the culchless oyster is less costly. However, once transplanted, the culched oyster has a better chance of surviving, especially in waters with high siltation (Matthiessen 1970). Thus, culchless oysters (and clams) should be raised to larger sizes in the hatchery, potentially offsetting any other savings.

The preparation of culch is tedious, taking about 12 minutes to produce one litre of sorted, crushed shell. The prepared culch is then cleaned and spread on the bottom of the set tanks at densities of 200 to 350 ml/square foot.

Setting

Prior to setting, the larvae are transferred to eight 825-litre tanks filled with warm, filtered, aerated water at densities of 2 to 18 million animals per tank. Maintenance is similar to that for larvae; the same algae mix is used, although cleaning is less frequent (every 72 hours). The animals are fed 70 litres of algae/day/tank during the first week; this is increased to 140 litres by the second week.

Because the animals are delicate at this stage of their life, their transfer too early to the juvenile tanks could reduce the overall survival rate. The hatchery does not heat the juvenile tanks and must wait for the bay water to heat sufficiently by natural means before young spat can be transferred. However, if the animals remain too long in the set tanks, they cannot receive the large quantities of flowing water and algae necessary for rapid growth, nor can the tanks be used for subsequent batches. Animals as small as 300 microns and as large as 1.9 mm have been transferred out of the set tanks in the hatchery.

Juvenile Development

From the set tanks, the young spat are transferred into either the juvenile tanks or the upwell system, or they are sold. Most oyster and clam seed are sold at about 3 mm; smaller animals than this would experience too low of a survival rate.

The juvenile tank has a capacity of 26,500 litres. The culched oysters are placed in trays that measure 0.5 x 0.6 m; these trays are then placed on the bottom of the juvenile tank. The tank is filled with enough sieved bay water to cover the trays and is aerated. Approximately 75 litres of algae are added per day to achieve a cell count of 2×10^6 cells/ml. The tank is cleaned twice per week, at which time the animals are washed and culled for death, fouling, and deformation. By the time the animals are ready for sale or transfer to the field, there has been a significant reduction in their numbers.

The upwell system is used only for clams and culchless or micro-culched oysters. The spat are held in screened-bottomed buckets, which are suspended over 825-litre rectangular tanks. Water from the bay holding tank is siphoned into these tanks, creating a continuous flow of water that aerates the spat. This sieved water is at bay temperature: 17^{o}C in May and 28^{o}C in August. Algae from the holding tank are added at a rate of about 7,600 litres per day. Animals have been kept in this system until they have reached 2.3 to 8 mm in diameter, and then they have been sold or transferred to the field.

The Hatchery's Physical Systems

The facility was designed for efficiency and flexibility to allow for changes in production techniques in response to new scientific information, environmental conditions or market changes. Simple changes in internal arrangements, made possible by movable tanks that can be used for a variety of functions, can streamline production. The hatchery was designed to facilitate easy cleaning. Energy, water and air systems were coordinated to maximize efficiency.

Energy System

The amount and the type of energy used in the hatchery depend on the production method and on the prevailing environmental conditions. Natural lighting and heating are used wherever possible. Because the bay water temperature is cool, the hatchery must heat large quantities of water to fill the brood, larvae and set tanks.

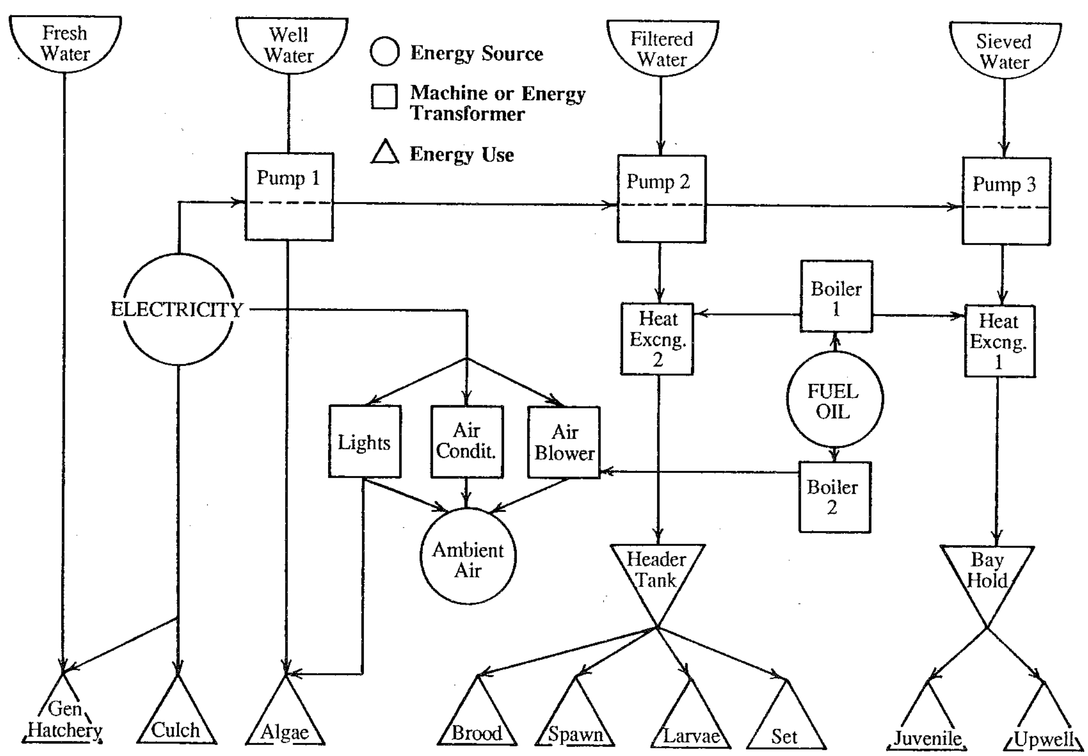
The energy flow is complex (Figure 6). Heat inputs include electricity for lighting and machinery and fuel oil for running the boilers to heat water and ambient air. Cooling inputs include air conditioners, bay water and ambient air. The relative contribution required from each of these sources to maintain ideal water and air temperatures is a function of the season. It is possible to determine the amount of electricity and fuel oil required if initial temperatures, final temperatures, and heating and cooling capacities are known.

Water System

Water is used in all phases of production: for all animal stages, for algae production and for cleaning. Water, the quality of which is critical to all hatchery operations (Bardach *et al.* 1972), can be sterilized with ultraviolet light or with chemicals. Fresh water for cleaning is obtained from a reservoir at no cost to the hatchery, and three pumps bring water from the bay (Figure 6). The first pump is for deep-well water, which is used exclusively for algae production. It is the purest of the three bay waters and is never heated. Pump 2 filters out particles greater than 1 to 5 microns; this water is used for larvae, brood and set tanks. The third pump delivers water to the juvenile room and sieves out particles greater than 700 microns.

The facility is designed to minimize the electrical costs of pumping. By using a series of different levels, water can be siphoned for various uses once it is within the

FIGURE 6. ENERGY AND WATER FLOW WITHIN THE HATCHERY



facility. Gravity is also used for the transfer of algae from the secondary to the tertiary stages and from the tertiary to the solar stage.

Air Flow System

If water is not kept in constant flow, the hatchery must provide aeration. Although the hatchery's air system is not elaborate, it is essential to algae and animal growth. Depending on size, all tanks are provided with a certain number of air tubes, except for the upwell system, which receives a constant flow of water. Air is pumped through these tubes from an air compressor that operates 24 hours per day. To reduce electricity consumption, the hatchery replaced its 5-HP air compressor with a 1-HP compressor, which has proven sufficient for the hatchery's needs.

Costs and Input Considerations

Before the decision to build a hatchery is made, a complete financial assessment should be conducted, including estimates of the initial investment cost, operating expenses and expected earnings. To date, existing hatcheries were estimated to have an average initial capital investment for building and equipment greater than $100,000 (Matthiessen and Smith 1979), but technically-advanced hatcheries, such as this one, may cost even more. Using land with direct access to the ocean can add considerably to the total price, although in this case, the land was already owned.

Hatchery operating costs were estimated at between $2,700 and $10,000 per month for most facilities (Matthiessen and Smith 1979). The actual amount depends on the hatchery's output and design, on the existing environmental conditions, and on unforeseen production risk beyond the control of the operator.

To date, there are only a handful of published estimates on the cost of bivalve seed, and the average costs per animal range from $0.0005 (Im and Langmo 1977) to $0.01 (Matthiessen 1979) for smaller seed and from $.029 to $0.25 (Gates *et al.* 1974) for larger seed. The most comprehensive estimates have been made in recent years by Im and Langmo (1977), Im *et al.* (1976) and Bolton (1982). They have attempted to estimate costs at each stage of production (e.g., brood, larvae etc.) using an economic engineering approach based on small, experimental hatcheries both on the Pacific and Atlantic coasts.

These studies are an important first step, but they fail to determine the cost of producing animals to different sizes in the same stage. This information is particularly critical at the final stage, that is, just prior to sale or transfer to outgrow facilities, because an aquaculturist purchases seed from a hatchery at a size in accordance with his production method, locality and production goals. It appears that 3 mm is a common size at which to sell the spat. Those who purchase larger seed face significantly less risk of mortality in open waters since they are more resilient to predation, siltation and climatic changes (Matthiessen and Smith 1979).

When larger seed are produced, it is usually because a) the hatchery cannot find buyers for seed not previously contracted for sale (usually the hatchery overproduces some animals to leave a margin of error), b) environmental conditions are such that the hatchery cannot transfer seed to their own "growout" facility or c) it is experimenting with growth techniques for larger animals. In other words, production to a larger size is not necessarily desirable but may be preferable to disposal of the animals if the hatchery can cover its costs at the margin.

Classifying Productive Inputs

An important first step in estimating the cost of raising bivalve seed to different sizes is to identify the input requirements for each stage of production. As outlined in Table 3, production of bivalve seed can be separated into five production stages and two support stages: 1) broodstock conditioning, 2) spawning, 3) larval development, 4) setting, and 5) juvenile development; and 1) culch and 2) algae preparation. Because input quantities change over time, they will also change over the duration of most stages. Therefore, in this analysis, the stages are broken down into smaller units of time, or periods. At the end of a period, there is an associated size and age

Table 3. Inputs to Bivalve Seed Production in the Hatchery

	Stages[a]								
	Production						Support		
	1	2	3	4	5		1	2	
Inputs	Condition	Spawn	Larvae	Set	Juvenile	Upwell	Culch	Algae	General Hatchery
ELECTRICITY									
Water pumping									
Wellwater									x
Filtered water	x	x	x	x					
Sieved water	x				x	x			
Air compressor	x	x	x	x	x	x		x	
Air cooling									
Lab room								x	
Algae room								x	
Air heating									
Office									x
Algae room								x	x
Larvae room		x	x	x					x
Back room	x			x					x
Equipment									
Office									x
Laboratory									x
Autoclave									x
Crusher						x			
Sorter						x			
Lighting									
Office									x
Laboratory								x	x
Algae room								x	
Larvae room								x	x
Back room									x
Juvenile room								x	x
FUEL OIL									
Water heating									
Heat exch.	x	x	x	x					
Filtered water	x	x	x	x					
Sieved water					x	x			
Space heating									
Office									x
Algae room								x	
Larvae room		x	x	x					
Back room	x			x					
LABOR									
Tank maintenance	x	x	x	x	x	x		x	
Culling			x	x	x	x			
Feeding	x		x	x	x	x		x	
Transfers			x	x	x	x		x	
Cleaning							x		x
Management									x
Laboratory work	x	x	x	x	x	x		x	x
Equipment maintenance									x
MATERIALS									
Antibiotics								x	x
Cleaning							x	x	x
Glassware								x	x
Lightbulbs								x	x
Screening									x
Office supplies									x
Fresh water							x		x
Nutrient solutions								x	
ALGAE									
Direct-feed tanks	x		x	x					
Solar tanks				x	x	x			
From natural seawater					x	x			
CULCH									
Micro				x					
Large				x					
MISCELLANEOUS									
Travel and communications									x
Insurance									x
Permits					x	x			
Shipping	x	x					x		
Loan payments									x
Taxes									x

[a]See text and Figure 5 for identification of stages. General hatchery inputs are independent of stage. Intermediate inputs are produced in support stages.

of animal, as determined by the growth rate experienced in the hatchery. The cost per animal at any age can be estimated by measuring the inputs that are used in a period.

As with most production systems, inputs can be classified as either fixed or variable. Fixed inputs are those whose quantity cannot be changed during the length of time under consideration since the cost of quick variation in their amount is large as to make such variations impractical (Mansfield 1979). Here, we are interested in smaller seed (produced in less than 4 months) as well as in larger animals that must be left in the hatchery longer. For convenience, the length of time under consideration is taken to be one year. Thus, fixed costs are assumed to include depreciation on building and equipment, taxes, and rental charges.

Some inputs are also fixed once a batch is started and so either do not vary with output or vary insignificantly. These inputs include general hatchery maintenance, such as for machinery and general hatchery cleaning. It is difficult to isolate these costs in terms of their applicability to any particular stage or period of the bivalve's life.

Variable costs differ in how much they are affected by the level of production output. First, because of the nature of hatchery production, which uses a modular-style (e.g., separate tanks), many inputs are only secondarily related to the number of animals within the system. For example, a larvae tank requires a certain amount of labor for cleaning and a certain amount of heated water no matter how many animals are in each tank even though the number of tanks used does depend on the total number of animals. Second, some inputs are related to the area where the animals are located and must be used no matter how many tanks are in that area, such as space heating. Third, some events take place only once and are not significantly affected by the number of animals, for example, sampling. Finally, there are those inputs, such as algae consumption, that vary directly with output.

It is the inputs that vary according to the number of animals in the system that are of greatest interest to this analysis, since it is these inputs that vary most with animal size. Inputs that do not vary directly with output are shared according to the relative length of time that the animal is retained in the hatchery, rather than the total output or the animal's size.

For this study, inputs are divided into six primary categories: electricity, fuel oil, labor, materials, algae and culch (Table 3). Although both algae and culch are support stages in the system, they are intermediate inputs that could be produced outside the hatchery and purchased.

In most hatcheries, it is not difficult to separate the total monthly operating costs into the total cost of each primary input. The real difficulty is in allocating a share of these primary input costs to different stages of production. Im, Johnston and Langmo (1976) made some progress on such a breakdown in costs. A summary of their results is presented in Table 4. These data highlight the importance of algae production as a component in the variable costs of bivalve production and serve as a benchmark by which to calibrate the results from the bivalve cost simulation program described in the next section.

IV - SIMULATING BIVALVE PRODUCTION COSTS

This section describes a simulation model by which one may calculate input requirements, net output and the cost of growing bivalve seed. Within the model, primary input costs are allocated to different stages of production, thus facilitating the comparison of the marginal cost of growing bivalve seed to a larger size with the expected marginal revenue of doing so. The model was designed to facilitate its coding as an interactive computer program that could be used by hatchery operators.

Structure of the Model

The model can be used to calculate production costs, inputs and output for an entire batch of bivalves. Even though most hatcheries would have concurrent batches at different stages of development, focusing on a single batch makes it possible to allocate input costs to the different production stages and to generate the information necessary to determine the optimal timing of production. It is also assumed that only one production pathway is followed for each application of the model; that is, all animals in all stages are treated similarly regarding the choice of culch, timing of transfers and all other aspects of the production processes.

To compute inputs and costs for animals of different ages, each batch is initially divided into the production stages discussed above. These stages are subdivided into shorter periods so that costs can be determined at various points in time within a stage. This is particularly useful for long stages, such as the juvenile stage, where a more precise estimate of the change in cost with age is required. All periods within a

Table 4. Production Costs in Various Production Stages in Bivalve Hatcheries of Two Sizes[a]

Stage	Percentage of Total Cost		Percentage of Variable Cost	
	Plant 1	Plant 2	Plant 1	Plant 2
Conditioning	2.7	0.2	3.9	0.2
Spawning	5.0	0.3	8.7	0.3
Larvae Rearing	15.0	27.8	21.4	25.1
Larvae Setting	7.2	18.8	7.6	13.5
Algae Production	38.9	33.3	29.4	39.0
Culch Preparation	3.7	9.3	7.2	14.7
Miscellaneous	27.5	10.3	21.8	7.1

[a] The source of the data is Im, Johnston and Langmo (1976). Plant 1 is an experimental plant with an output of 15 bushels/week. Plant 2 is scaled up to a projected 800 bushels/week.

stage are of equal length, but the length and number of periods may differ amongst stages.

The number of bivalves surviving and their size are calculated at the end of each period, along with the per day averages over the period. The quantities and costs are calculated for each period and are averaged over each period as well. The costs incurred in the current period are added to those of previous periods to accumulate total costs up to the end of the period. This type of structure, in which there is a succession of repeated computations, lends itself easily to FORTRAN, the computer language in which the model is coded.[4]

As shown in Figure 7, the options to run the code are chosen in *step 1a*, and data, including price of inputs, are entered or computed in *steps 1b-1g*. After all data are entered, a batch is begun on the first day of the first month of production *(step 2)*. Each stage *i*, beginning with the conditioning stage, is looped through *steps 3-14*, and within this stage loop, period *j* is incremented *(steps 5-13)*. The passage of time during the course of a batch is accumulated within j *(step 6)*, and thus the age of the bivalves is advanced.

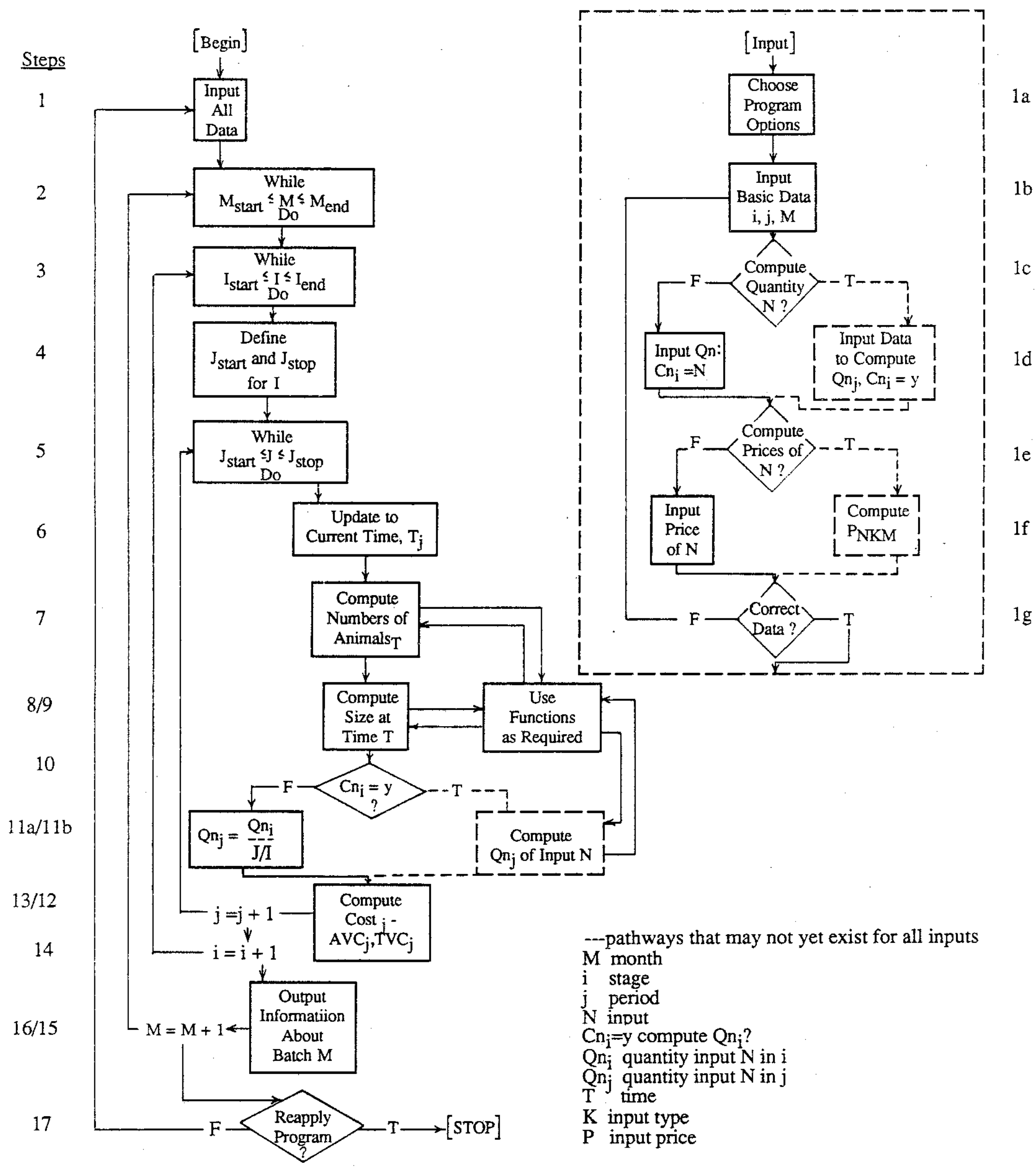
[4] Preliminary development of the code, which concentrated on specific subroutines, is shown in Myers (1985). A listing of the completed code is given in Appendix A. It is being run on an IBM mainframe computer. Data can be entered entirely by file, entirely interactively, or by a combination of both.

With this nested looping structure, any computation within the *j* loop will depend not only on *j*, but also on time, stage and the month the batch is begun.

Survival rates for bivalves *(step 7)*, size of bivalves *(step 8)* and input quantities *(step 11)* are computed within the loop j, calling function subroutines, which are placed outside of the loop when used by more than one subroutine. Cumulative and average costs, based on information from previous steps, then are computed *(step 12)*. The program output contains information generated about the batch beginning at month *M (step 15)*; a new batch is begun on the first day of month *M+1 (step 16)*. By performing computations for each month, information about batches begun at different times during the production season can be generated.

The options included in the code allow the model to be adapted to fit a variety of different hatcheries. With little effort, the internal parameters of the code can be respecified to change the number, units and name of inputs, and the number and name of stages, thus adapting the code to a hatchery's specific production methods. When running the program, the user may specify the length of each stage and the number of periods within that stage, thus determining in part the level of detail contained in the information generated. It is possible to skip entirely stages for which information is not required or for which data may be insufficient.

FIGURE 7. BIVALVE CODE FLOWCHART



The user may also choose the way in which input prices and quantities are determined. If the code contains the necessary algorithms, the user may choose either to allow the program to compute input price and/or quantities *(steps 1f, 11b)* or enter these as fixed parameters *(steps 1d,1f)*. The modular design of the code facilitates the addition or deletion of these price and quantity algorithms.

Algorithms

The algorithms for bivalve survival rates, bivalve size and production costs are required in the model and are described here.[5] In addition, the optional algorithms for algae price and quantity, which constitute the major focus of this analysis, are outlined. These algorithms, except those for price, are contained within the period loop and thus depend on time, period and stage. The prices for inputs used in a period are functions of the elapsed time up to period *j* and on the month during which that period occurs.

Survival Rate

The output from a batch of bivalves, as well as the quantity of inputs and production costs, depends on the initial number of bivalves in the batch and on the number of animals surviving from period to period. Death of a proportion of the population in a hatchery is unavoidable even though the mortality rate is lower than in the wild. Survival rates are affected by environmental conditions maintained in the tanks, but little is known about the trade-off between increasing the survival rates and cost of improving the conditions in the tanks. Survival rates will also vary by hatchery, although in all hatcheries survival should increase at an increasing rate because the animals become more resilient with age and culling rates go down with increasing age.

Hatcheries often overproduce at early stages to account for the declining numbers of animals over time, which is a result of natural mortality and the culling of inferior animals from the batch. Culling assures a faster overall growth rate and a superior final harvest. However, it is difficult to distinguish between the effects of mortality and culling on the number of remaining animals because both dead and inferior animals are removed in the same sorting process.

It is reasonable to assume that the number will decline over time. There are, however, few, if any, data on actual survival rates under these controlled conditions. At the time data were collected for this study, the hatchery had not been in production long enough to have reliable data on survival rates. For this reason, a general exponential function reflecting a constant percentage mortality rate over time was written into the code. This allows the user to specify particular survival rates and facilitates testing the sensitivity of costs to changes in survival rates.

The number surviving at the end of each period is computed at *step 7* of the code by calling a function that relates survival at time *t* to the number of animals initially in the batch, i.e., the number successfully spawned.[6] This function is:

(1) $N_t = N_o(t)^{\rho}$,

where N_t = the number of animals remaining at time *t*; N_o = initial number of animals spawned; and ρ = a parameter to be set by the user, $0 > \rho > -1$.

While this general functional relationship is thought to be an accurate representation of an invertebrate population's survival rate (Emmel, 1976, p. 205), its tendency to go to infinity as *t* goes to zero caused a problem in writing the computer code. To be consistent with the use of *t* as an index of the passage of time for other calculations, it was necessary to have equation (1) at $t=0$ record the initial number of animals in the batch (N_o) Thus for convenience of programming, N_t was defined as the number of animals remaining at the end of any week *t*, where *t* is incremented in discrete time. Thus, to reconcile the initial estimate of the number of bivalves with the continuous time function above and the other functions in the model, the following function is used:

(2) $N_t = N_o\,(t+1)^{\rho}$.

[5] The specific forms of the mathematical relationships used to describe the algorithms were chosen primarily because they were found to provide the "best" statistical results given the data used to estimate the parameters of the functions empirically. The empirical estimates are described in subsequent sections.

[6] Although the cost to condition the adult population (stage 1), which spawns the initial egg population, affects the total cost of producing a batch, the number of adults is not important in determining the initial population in this model, because N_o is input by the user as a fixed parameter rather than as a function of the fecundity of the adults spawned.

An estimate of the average number of animals surviving during a period is necessary to calculate an average quantity of inputs used during j. The actual average number over the period from time a to b is calculated by:

$$(3)\quad N_{(a,b)} = \frac{N_0}{(b-a)} \int_a^b (t+1)^{\rho}\, dt.$$

This integral was estimated in the code with a Reiman sum using the trapezoidal approximation to the area under a curve. The method gives a satisfactory estimate of the integral and increases the generalizability of the code since it does not depend on the actual form of the integrand.[7]

Size

Both the size of an individual bivalve at the end of each period and the average size during the period are estimated. This information is useful in assessing potential market price and is also necessary for computing the amount of inputs used within a period.

Data relating the size of bivalves to their age were difficult to find in the literature as well. However, Epifanio, Logan and Turk (1976) and Epifanio (1975) report a small amount of data relating the size H_t (in mm of shell height) to age in weeks (t) for both clams and oysters. These data suggest that the relationship is quadratic, at least for young bivalves:

$$(4)\quad H_t = \alpha t + \beta t^2,$$

where one would expect $\alpha \geq 0$ and $\beta < 0$, (the parameters are chosen by the user). The only potential problem with an empirical specification of this function is that it may yield a maximum size at too early an age. This problem is not serious because the function behaves well over ages that are likely to exist in a hatchery setting.

Using this relationship, the average size during any period starting at time a and ending at b is given by:

$$(5)\quad AH_{(a,b)} = \frac{1}{(b-a)} \int_a^b (\alpha t + \beta t^2)\, dt.$$

This integral also is approximated in the code using the trapezoidal rule, which computes the value of the function at each day of the period.

Costs

Estimating the variable cost to produce a single bivalve to a certain age or size involves calculating the total variable cost (TVC_t) for an entire batch at some time t and dividing by N_t, the number of animals surviving is given by:

$$(6)\quad AVC_t = TVC_t / N_t.$$

This average variable cost per animal is derived at the end of each period.[8]

As stated above, each batch is initially divided into stages $(i=1,\ldots,S)$, which are in turn divided into periods. To facilitate cod-

---

7 To estimate the definite integral $\int_a^b f(t)dt$ by the trapezoidal rule, one must calculate

$$A = \int_a^b f(t)dt \doteq \sum_{k=1}^{n-1} f(t_k)\ \Delta t + \frac{f(a) + f(b)}{2}\ \Delta t.$$

For coding purposes, it was efficient to calculate

$$A = \sum_{k=1}^{n} f(t_k)\ \Delta t + \frac{f(a) - f(b)}{2}\ \Delta t$$

and $\Delta t = 1/7$, a daily increment. If f is continuous on $a \leq t \leq b$ and twice differentiable on $a < t < b$, then there is a number $a < \epsilon < b$ such that

$$\int_a^b f(t)dt = A - \frac{(b-a)}{12}\ f''(\epsilon)\Delta t^2$$

There is no set procedure for finding the value of e such that this is true, but one can obtain some idea of the magnitude of the error by investigating the size of f"(t) between a and b (Thomas, 1960, pp. 207-17 and pp. 385-88).

8 The stage associated with the period must be greater than 1 since no animals are produced in the conditioning stage.

ing, and allow for a different number of periods in each stage, periods $(j=1,...,J)$ are indexed consecutively over the entire batch. Thus total variable costs at time t^* (measured in weeks) is the sum of all variable costs $(PTVC_j)$ across all periods included in the first t^* weeks:

$$(7)\quad TVC_{t^*} = \sum_{j \in t^*} PTVC_j.$$

Because the code keeps track of periods and stages by incrementing the *jth* loop within the *ith* loop, costs up to the end of a period may also be thought of as the sum of the costs in all periods included in the completed stages $(i=1,...,S^*-1)$ plus the sum of the costs for all completed periods up to time t^* included in the current stage, S^*,

$$(8)\quad TVC_{t^*} = \sum_{j \in (S=1,...,S^*-1)} PTVC_j \quad + \sum_{j \in (S^* \cap t \leq t^*)} PTVC_j\,.$$

Variable costs during a period are of course determined by the quantity of the inputs used during that period and by the price of the input during the month in which the input is used. This is incorporated into the model as:

$$(9)\quad PTVC_j = \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{r=1}^{R} P_{nkm_j} Q_{nkrj},$$

where P_{nkm_j} = price per unit of input n, type k, in the month during which j occurs; Q_{nkrj} = quantity of input n, type k, for use r used during j; N = number of inputs n; K = number of types k of input n; R = number of uses of input n, type k; and m_j = month during which j occurs.

<u>Prices and Quantities</u>

As discussed previously, prices and quantities are either computed by algorithms within the code or are input by the user, thus allowing the user to focus on specific parts in the production process. Although the model allows any of the input prices to be computed, this option is probably most useful in computing the cost, or internal price, of algae and culch, because these two inputs are often intermediate inputs produced within the hatchery. This analysis focuses on the computation of this internal price for algae and on the use of this input as feed during the production of the bivalves.

As shown in equation (9), the model allows for a price for K different types of input n in each month of production, thus creating a 3-dimensional price array, P_{nkm}. The code requires a value for each element of the array regardless of whether the price is computed or input by the user so that a corresponding price can be found for inputs used during any month.

From the computations for the quantity of inputs used in each period, where a quantity is calculated for each type of each input for each period, Q_{nkj}, another 3-dimensional array is constructed. Determining the quantity of each type involves summing over the R different uses for the input. For expository purposes, this is included as an additional subscript on the quantity variable in equation (9), Q_{nkrj}. However, in the code, this sum of input quantities by use is calculated in the quantity subroutine, and this subscript does not appear in the quantity array used in the cost calculations.

If the quantity of input is entered by the user rather than computed by algorithms in the code *(step 1d)*, a lump sum is entered as Q_{nki}, which is then divided equally among each period of that stage *(step 11)*.

Algae Price and Quantity

The decision to focus this report on algae production (price) and on the quantity of algae used for feed in bivalve production is justified for three reasons.

First, the production of algae in the hatchery was the most well defined and firmly established of the hatchery's production processes. It was easy to separate the inputs used in the production of algae from those used as direct inputs to bivalve production, because some different types were used and because the two processes were physically separated in the hatchery.

Second, the algae production model closely resembles the bivalve code structure because of the similarity in the production processes. By first writing the algae price code, potential problems with the full bivalve code could be anticipated.[9]

Third, and most important, is the fact that feed costs have been identified as the largest single component of hatchery costs, ranging from 29% and 33% of variable costs in early studies (Im *et al.* 1976) to as high as 85% in more recent studies (Bolton 1982). Thus, from a practical point of view,

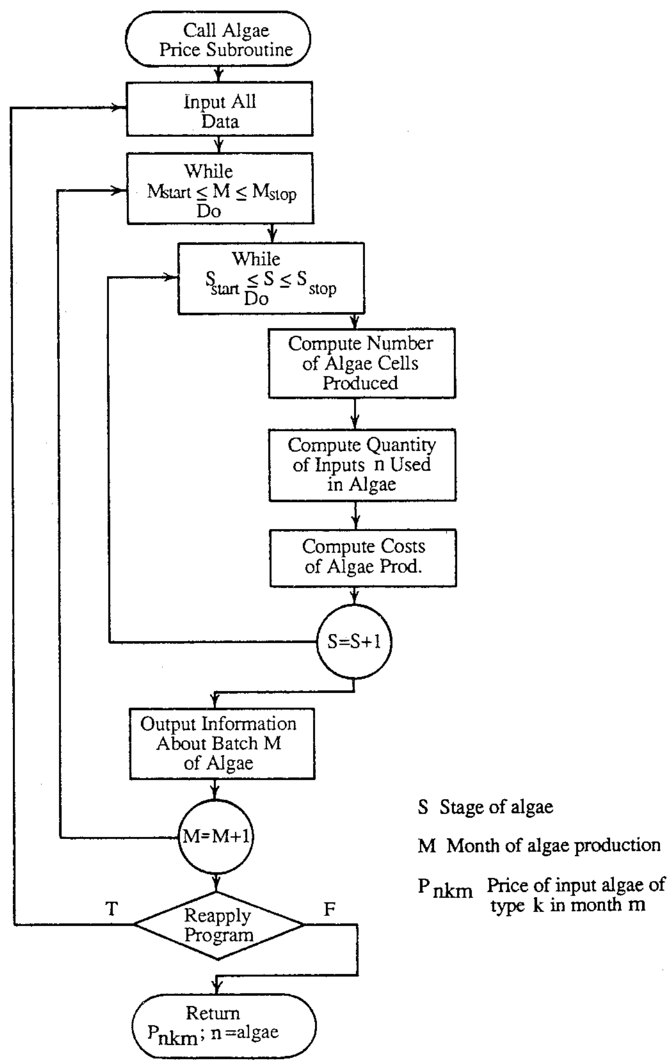
> ... the major bottleneck to commercialization of intensive bivalve culture systems [lies] in the ability to economically culture massive quantities of these suitable algal species. Intensive study of problems associated with such mass culture of algae is certainly warranted... (Epifanio 1975, p. 190).

These concerns become even more important as hatcheries attempt to raise animals to a larger size.

The algae system, described in Section *III*, can be broken into stages of relatively short, fixed duration, thus distinguishing different algae "types" by the age of the culture. Algae can be fed to the bivalves from different stages and as such provides one example of how different types of one input can be used within the code. These different types of algae are also distinguished on the basis of their different physical characteristics, such as density, and overall purity, and by the cost of production. In the subroutine where algae costs are computed, a price, which in this case equals production costs, is computed for each of the different algae types.

Figure 8 is a representation of the algorithms used to simulate algae production. As in the bivalve program, only one batch of algae is followed at a time. It is also assumed that only one algae species, or a single mixture of species, is produced in a batch and that all tanks within a stage are used. By assuming that a new batch is begun on the first

FIGURE 8. ALGAE PRICE SUBROUTINE



[9] In fact, the two processes are very similar inasmuch as both operate on a batch-like principle. The transfer of algae to successively larger tanks as they mature is comparable to the transfer of animals between stages. Both systems are a function of the number of animals or cells; however, the number of organisms increases over time in the algae system due to cell reproduction, although cell size remains constant.

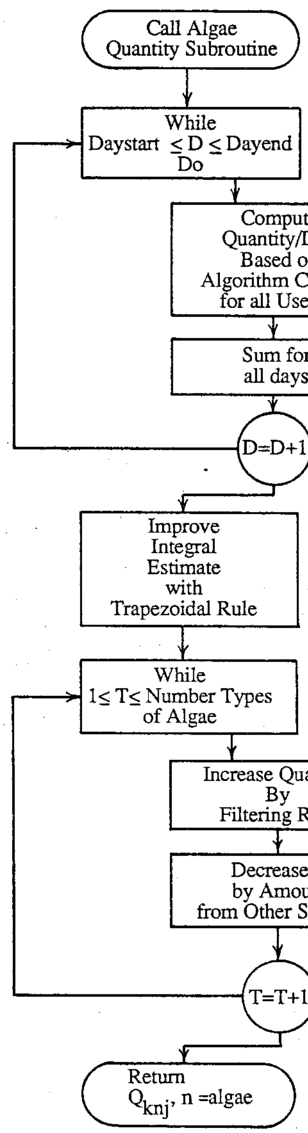
of every month, one can derive the 3-dimensional price array required by the bivalve program, P_{nkm_j} from equation (9) (where n = algae), from the internal cost of producing each type of algae.

The cost of producing any stage of algae depends on the amount of inputs --electricity, labor and materials-- used during the stage and on the price of those inputs during the month in which the stage occurs. Because algae production in the hatchery is separate from bivalve production, there is a provision in the data input to specify different prices for those inputs used in producing algae from those used in other parts of bivalve production. The algorithms used to derive the internal costs of producing algae are discussed in detail in Myers (1985).

The algae quantity subprogram is used to compute the feed requirements for a batch of bivalves at any time t (see Figure 9). It allows for the feeding of algae from any stage of maturity to bivalves at any stage of growth, although it is likely that average production costs per unit of algae would be too high in the early stages. Exactly at what stage during the algae production process algae should start to be used for feed is an empirical question that can only be answered by the information generated from the program itself. More is said about this issue below.

To estimate the algae required for a batch of bivalves, one must know the relationship between feed consumption and some measure of a bivalve's age or size, in addition to the number of animals remaining in the batch at time t. In the literature, one finds feed requirements related directly to age (Epifanio 1975) and directly to size, as measured by shell height or dry weight (Epifanio *et al.* 1975 and Bolton 1982). To maintain the flexibility of the program, both options were included in the code. This was accommodated easily, because regardless of the relationship used --size and feed or age and feed-- the data are consistent with a function that is linear in logarithms. One might well interpret the relationship as a Cobb-Douglas production function.

FIGURE 9. ALGAE QUANTITY SUBROUTINE



In the first option contained in the code, the daily consumption of algae (10,000 cell units) at time t, C_t, is related to shell size in mm of shell height as determined from equation (4). Thus,

(10) $ln(H_t) = \delta + \gamma \; ln \; C_t.$

Solving for C_t and substituting,

(11) $C_t = (e)^{-(\delta/\gamma)} (H_t)^{1/\gamma}.$

Multiplying this expression by the number of animals surviving (N_t) and substituting from equations (7) and (9), the total daily feed requirements (in 10,000 cell units) for a batch at time t, C_t^* can be written entirely in terms of t:

$$(12)\quad C_t^* = [(\alpha t + \beta t^2)^{1/\gamma}]\,[(t+1)^{\rho}\,N_0]\,[e^{-\delta/\gamma}].$$

The total number of algae cells consumed (in 10,000 cell units) during a period is then given by

$$(13)\quad \overline{C_t} = e^{-\delta/\gamma}\int_a^b [\alpha t + \beta t^2]^{1/\gamma}[(t+1)^{\rho}]dt.$$

Again, the values of the parameters are left to the user.

The second option relates the cells consumed, c_t, directly to age in weeks, thereby eliminating the need to evaluate equation (4). The new relationship is[10]

$$(14)\quad \ln t = \xi + \eta \ln c_t.$$

Thus,

$$(15)\quad c_t = (e)^{-(\xi/\eta)}\, t^{1/\eta};$$

$$(16)\quad c_t^* = [t^{1/\eta}]\,[(t+1)^{\rho}\;N_0\,]\,[e^{-\xi/\eta}];$$

and

$$(17)\quad \overline{c_t} = e^{-\xi/\eta}\int_a^b [t^{1/\eta}\,]\,[(t+1)^{\rho}\,N_0\,]dt.$$

Both of these options create the 2-dimensional array for algae quantity.[11] The price determined in the algae price routines can then be referenced to compute the cost of the algae inputs used in each period of bivalve production.

10 The variables C_t and c_t refer to feed consumption (in 10,000 cell units), calculated according to options 1 and 2, respectively.

11 Both equations (13) and (17) are evaluated in the program using procedures similar to those for the survival and size equations above and reference equations (2) and (4) for each day of the period.

V - EMPIRICAL TESTS OF THE SIMULATION MODEL

The purpose of this section is to describe the results from initial experimentation with the bivalve cost simulation model. The decision to apply the model to oyster seed production (*Crassotrea virginica*) was based primarily on the availability of data. As emphasized above, the empirical results reported here focus on simulating the cost, or internal price, of algae and on the quantities of algae and total feed costs of raising a batch of bivalves to various sizes. The remainder of this section begins with a discussion of the data used for the analysis. Next, the empirical results are presented for a base case, after which a variety of sensitivity analyses are conducted to see how the empirical results are affected by changes in the key parameters of the model.[12] Finally, these results are compared with other cost and bivalve seed price information to provide initial indications of the optimum size at which to market the seed.

The Data

To experiment with the model, it was necessary to obtain two sets of information: a) the data required by the subprogram that calculates the cost or internal price of algae and b) data by which to estimate the parameters of the survival function, the age-size relationships and the algae feed functions [equations (1), (4), (10) and (14)]. The data for the algae price (internal cost) estimates are based in large measure on several weeks of observation at the hatchery during the summer of 1983. There were, however, a few items taken from other sources, but these were modified to be consistent with the other production processes and specific environmental conditions of the hatchery. These data are summarized in Appendix B and are described in detail by Myers (1985).

For any single type of bivalve, there were very few data in the literature on which to base the estimates of the parameters for

12 Myers (1985) conducts considerable sensitivity analysis on the internal price of algae, so these results focus on changes in other parameters of the system.

equations (1), (4), (10) and (14). However, except for the survival equation, it was possible to combine data from several sources to derive consistent parameter estimates for the oyster.

Table 5 contains an empirical estimate of equation (4a), relating the size of oysters to age. The data are from Epifanio *et al.* (1976).

Within the simulation model, there are two alternatives for estimating the feed (algae) requirements for bivalves. One way is to relate feed directly to shell height as in equations (10a,b) of Table 5. To estimate equation (10a), equation (4a) was used first to predict shell height at some selected ages for which Claus and Adler (1970) report algae consumption data. These predicted shell heights are then regressed on the algae consumption figures (in log-linear form) to obtain equation (10a). Equation (10b) was estimated directly from data in Bolton (1982). A second way to estimate feed requirements accommodated in the model was to relate algae consumption directly to age. Data from Claus and Adler (1970) were used to estimate equation (14a) directly, while equation (14b) was estimated directly from information in Bolton (1982).[13]

The primary reason for developing these four ways of estimating feed requirements was to test the sensitivity of the model. It is somewhat surprising, but encouraging, to find that the algae production elasticities in these four equations are similar, ranging only from 0.35 to 0.42.

Data with which to estimate the survival rate parameters were among the most difficult to obtain. There were no population survival data for oysters raised in a hatchery, although Bolton (1982) makes allowance for a 20% mortality rate during the cultivation period. Because most assumptions in Bolton's experiment proved extremely optimistic, it seemed inadvisable to use this overall mortal-

Table 5. Estimated Equations for Use in Simulations

Equation Number[a]		Estimated Equation	
(1a)[b]	$ln\ N_t =$	8.55 -	0.255 $ln\ t$
		(46.09)	(-6.26)
	$R^2 =$	0.66	[0.041]
(4a)[c,d]			
	$H_t =$	0.737(t) -	0.00252$(t)^2$
		(22.14)	(-6.99)
		[0.033]	[0.0004]
(10) (a)[f]	$ln\ H_t =$	-0.631 +	0.338 $ln\ C_t$
		(-1.65)	(4.58)
	$R^2 =$	0.81	[0.074]
(b)[e]	$ln\ H_t =$	-1.769 +	0.416 $ln\ C_t$
		(-5.85)	(18.57)
	$R^2 =$	0.98	[0.022]
(14) (a)[e]	$ln\ t =$	-0.331 +	0.345 $ln\ C_t$
		(-0.85)	(4.62)
	$R^2 =$	0.81	[0.075]
(b)[d]	$ln\ t =$	-2.244 +	0.407 $ln\ C_t$
		(-5.644)	(13.83)
	$R^2 =$	0.97	[0.029]

[a] See text for equation numbers and variable definitions. The numbers in parentheses and brackets are t-ratios and standard errors, respectively.

[b] The source of the data is Eldridge, et al. (1979).

[c] R^2's are not applicable for restricted regressions.

[d] The source of the data is Epifanio et al. (1976).

[e] The source of the data is Bolton (1982).

[f] The sources of data are Claus and Adler (1970) and Epifanio et al. (1976).

[13] Bolton (1982) estimates oyster feed requirements through a mathematical relationship between the whole weight of oysters and the daily consumption of algae (due to Pruder et al. (1977)):

$$Y = 8.2\ x^{-0.21}$$

where x = whole weight in grams; and Y = algal cells cleared x 10^8/g whole weight/day. Because they report data on age and shell height for oysters of different whole weights, once this equation is used to estimate algal consumption based on weight, these predicted values can in turn be regressed on age and shell height to obtain equations (10b) and (14b).

ity rate as a base of comparison. Furthermore, there was no easy way to infer from this overall rate how mortality changes over time.

Since hatchery data on which to estimate survival rates of oysters or other bivalves were unavailable, the best alternative to obtain survival rates that are at least in the relevant range was to estimate equation (1) from data for hard clams planted in protective trays in a South Carolina estuary (Eldridge *et al.* 1979). The size of the clams planted was larger than bivalves raised in a hatchery, and the survival function for smaller bivalves had to be extrapolated beyond the limits of the data. Initially, one might except this to lead to an overestimate of the survival of hatchery-sized bivalves. However, the data do reflect a less controlled environment than found in the hatchery, a factor that may partially offset the effect of size. The estimated equation, in log-linear form, is in Table 5; its parameters imply a constant mortality rate of 25.5%. When accumulated on a weekly basis, this function leads to over 60% mortality in a year's time, three times that for oysters in Bolton's (1982) prototype hatchery. This discrepancy is large, but it is unlikely that Bolton anticipated a year-long cultivation period. Equation (1a) from Table 5 implies about a 50% mortality after 18 weeks. This is closer to Bolton's initial assumptions. These two extremes are a useful range over which to test the sensitivity of the results.

Empirical Results

A computer program was designed that implements the simulation model described in detail above. This program was designed to be used by the hatchery operator to generate information about a batch of bivalves at different states of development. Input costs and requirements, and bivalve age, survival and size are reported at the end of each production period and as cumulative amounts. Most important, it gives an average cost per bivalve at the end of each period to give the user information about production timing and about the expected revenue from bivalve seed sale that might be expected.

Initial Simulations

A partial listing of the printout generated by the program is in Appendix B. It contains information for a batch of oysters under base run conditions (run 1). The batch is assumed to begin in April and last for 420 days, or 59 periods. Myers' (1985) input and price data were used to calculate the internal cost per unit of algae.[14] The data reflect economic conditions at the hatchery in 1983. The batch contains one million oysters initially; the survival rate for bivalves is given by equation (1a), of Table 5; the age-size and feed-size relationships are given by equations (4) and (10a), respectively. The bivalve feeding filtering rate is assumed to be 50%, and 12% of the diet is assumed to come from algae in circulating sea water in the juvenile stage.

The assumptions underlying the base run and 11 other simulations are given in Table 6. These runs contain numerous combinations of the functional relationships described earlier in Table 5. The first three runs assume the same mortality rates but accommodate different ways of estimating the physical feed requirements. Runs 1 and 3 assume feed is directly related to size, although in run 2, feed is directly related to age. The results of these runs are summarized in Table 7.

For the base run, the number of bivalves surviving over the duration of the batch falls from a million at spawning to an estimated 938,000 at the beginning of the larval stage. After setting, the estimated number is 716,000. If the batch were to remain in the hatchery for 420 days (including 60 days for conditioning the broodstock), the

---

14 To facilitate comparisons, all other runs of the model are assumed to begin in April and run for 59 periods (420 days). The simulation was run over such a long duration primarily to test the behavior of the model since, due to bivalve size and the cost of feed, it is clear that bivalve seed would be sold or transferred to grow-out long before this last period was reached. Conditioning takes a month, and the spawning, larvae and setting stages were broken into periods of a few days each; but once the juvenile stage was reached, periods were a constant one week in length. For ease of exposition, much of the discussion in the text focuses on monthly intervals or on individual stages.

Table 6. Description of Parameters for Alternative Simulations Runs[a]

Simulation	Survival Equation	Age-Size Equation	Feed-Size Equation	Feed-Age Equation
Run 1 (Base Run)	$N_o = 1 \times 10^6$ $\rho = -.0255$	$\alpha = 0.737$ $\beta = -0.00252$	$\delta = -00.631$ $\gamma = 0.0338$	N.A.
Run 2	same as run 1	same as run 1	N.A.	$\xi = -0.331$ $\eta = 0.345$
Run 3	same as run 1	same as run 1	$\delta = -1.769$ $\gamma = 0.0416$	N.A.
Run 4	$N_o = 1 \times 10^6$ $\rho = -0.214$	same as run 1	same as run 1	N.A.
Run 5	$N_o = 1 \times 10^6$ $\rho = -0.296$	same as run 1	same as run 1	N.A.
Run 6	$N_o = 1 \times 10^6$ $\rho = -0.163$	same as run 1	same as run 1	N.A.
Run 7	$N_o = 1 \times 10^6$ $\rho = -0.337$	same as run 1	same as run 1	N.A.
Run 8	same as run 1	$\alpha = 0.671$ $\beta = -0.00252$	same as run 1	N.A.
Run 9	same as run 1	$\alpha = 0.671$ $\beta = -0.00332$	same as run 1	N.A.
Run 10	same as run 1	same as run 1	$\delta = -0.631$ $\gamma = 0.412$	N.A.
Run 11	same as run 1	same as run 1	$\delta = -0.631$ $\gamma = 0.486$	N.A.
Run 12	same as run 1	$\alpha = 0.671$ $\beta = -0.00332$	$\delta = -0.631$ $\gamma = 0.486$	N.A.

[a] See Table 5 and equations (1), (4), (10) and (15) in the text for parameter definitions and estimated relationships.

number of animals surviving would be approximately 357,000.

The estimated average size of the bivalves ranges from just under 2 mm after the setting stage to just over 33 mm at the end of the simulation. As the batch progresses through time, the additional cost of feeding larger bivalves is partially offset by declining numbers. Because the feed cost per surviving bivalve includes the accumulated costs of feeding those that did not survive, the costs rise rather rapidly throughout the batch. At the end of period 18 (about 100 days), the feed cost per animal is still less than one cent, but if left in the hatchery through the 59th period, feed costs would average more than a dollar per surviving bivalve.

Table 7. Summary of Simulation Output for Initial Runs

Period[a]	Stage	Cumulative Days to End of Period	Animals Surviving (thousands)	Average Size of Animal (mm)	Cumulative Cost per Bivalve: Run 1[b] (Base) ($)	Run 2 (Proportion of Run 1)	Run 3 (Proportion of Run 1)
6	Setting	49	716	1.98	c	0.98	8.77
10	Juvenile	77	594	4.84	c	0.97	5.72
14	Juvenile	105	534	7.61	0.003	0.98	4.29
18	Juvenile	133	495	10.30	0.009	0.99	3.59
22	Juvenile	161	467	12.91	0.022	1.01	3.16
26	Juvenile	189	446	15.44	0.045	1.04	2.86
30	Juvenile	217	429	17.89	0.082	1.06	2.63
34	Juvenile	245	414	20.26	0.137	1.09	2.46
38	Juvenile	273	402	22.55	0.214	1.12	2.31
42	Juvenile	301	391	24.76	0.316	1.16	2.20
46	Juvenile	329	382	26.88	0.449	1.19	2.10
50	Juvenile	357	373	28.93	0.616	1.23	2.01
54	Juvenile	385	366	30.90	0.822	1.27	1.94
59	Juvenile	420	357	33.24	1.144	1.32	1.86

[a]Period 6 is through setting (about 50 days including adult conditioning, spawning and larval stages); the remaining periods are a week in length.

[b]See Table 6 for assumptions about runs.

[c]Less than 0.001.

Data for the other runs in Table 7 suggest that these results can be quite sensitive to the assumptions made regarding the feed equations. In the base run, feed requirements are related directly to the size of the animal, whereas in run 2 it is assumed that feed requirements are determined by equation (14a) relating feed directly to age. Assuming the same survival rates, this alternative way of calculating feed leads to a slightly lower estimate of feed cost per surviving animal through period 18, but then cost rises to 1.3 times that of the base run by period 59. The feed relationships in these two runs were estimated from the same basic data; despite these differences, the results are quite consistent.

The cumulative costs from run 3 are in sharp contrast to those for runs 1 and 2. In this latter run, costs are consistently higher than in the base case, and within the same run the differential is most extreme in the early periods. This result is somewhat surprising, given that the production elasticity for algae is higher in runs 3, 4 and 6 than in the base run. However, the intercepts on the size or age axes of the production function are lower than for the base run. Thus, even though the percentage gain in size is larger for each one-percent change in algae consumption, absolute feed requirements to raise these oysters to any size over the duration of the batch remains higher than in the base run.

The implications of this result lie in the importance of measuring feed requirements accurately in early stages of the growth process. That is, if one were to look only at a single animal, the tendency would be to conclude that feed costs would be relatively unimportant compared to costs for older animals. However, because the number of animals falls throughout the process, costs (due to what appear to be minor cost differences in early periods) accumulate rapidly as more animals die and as total costs are averaged over fewer survivors. Thus, reliabile estimates for both feed costs at different points in the growth process and the number of animals surviving are more critical in estimating costs for raising bivalves than for larger animals (e.g., cattle or other livestock) with much lower mortality rates.

A Digression on Algae Production

The primary focus of the analysis so far has been on the costs of feeding a batch of bivalves and the effect of survival and feeding ratios on costs. However, it is evident (Table 8) that cumulative algae consumption in runs 2 and 3 is a smaller multiple of the base run consumption than it is for cost in early periods and is a larger multiple in later periods. The primary reason for this is that feed costs for bivalves are also a function of the implicit price or cost of algae that is raised for feed. The program uses algorithms from Myers (1985) to calculate the price of algae, which are assumed to be constant in any month but which can vary by month because of seasonal differences.

The "algae price" algorithm assumes that algae are produced in a batch process, which is divided into 6 stages covering about 6 to 7 weeks. During the first four stages, most of the algae are needed to innoculate the subsequent stage. Thus, the net production of algae available is negligible until the tertiary stage (about 6 weeks). Up to this point the algae are grown in well water and the batch is quite pure, thus making it ideal for feeding to bivalves at the larval and set stages. These tertiary algae tanks are located (in the study's hatchery) adjacent to the larval tanks, which

Table 8. Algae Consumption for Selected Simulations

Period[a]	Cumulative Algae Consumption: Run 1[b] (Base Case) (10 bil. cells)	Run 2 (Proportion of Run 1)	Run 3 (Proportion of Run 1)	Ratio of Cost to Algae Consumption[c]: Run 2	Run 3
6	48	0.98	8.05	1.00	1.09
10	580	0.97	4.79	1.00	1.19
14	2,153	0.98	3.68	1.00	1.17
18	5,067	1.01	3.10	1.00	1.16
22	9,513	1.04	2.73	0.97	1.16
26	15,603	1.07	2.47	0.97	1.16
30	23,382	1.11	2.27	0.95	1.16
34	32,846	1.15	2.12	0.95	1.16
38	43,947	1.20	2.00	0.93	1.16
42	56,605	1.25	1.89	0.93	1.16
46	70,711	1.30	1.81	0.92	1.16
50	86,133	1.36	1.74	0.90	1.16
54	102,720	1.42	1.67	0.89	1.16
59	124,850	1.50	1.61	0.88	1.16

[a]See Table 7.

[b]See Table 7.

[c]Ratio of column 7 (Table 7) to column 3 (Table 8) and column 8 (Table 7) to column 4 (Table 8), respectively.

helps keep the costs of moving algae around the hatchery to a minimum. There are an estimated 9,650 litres of algae available at the tertiary stage, most of which is fed to the larvae and animals at set. Some of the tertiary algae is also used to innoculate a solar stage to which seawater, rather than well water, has been added, thus making it unusable for the sensitive larvae. After another week or two, there are an estimated 67,000 litres of algae available to feed juvenile bivalves. Because the cost per cell of algae in the solar stage is about 25% of that in the tertiary stage, it is not surprising that in the early periods feed cost is higher relative to algae consumption than it is in later periods (Table 8).

A further explanation for the behavior of costs relative to algae consumption lies in a slightly higher algae cost estimate in the summer months because of the air conditioning required to maintain cool temperatures suitable for algae production. For example, assuming the batch is started in April, about 82% of the cost of algae in the solar stage is due to labor, 5% to materials and 13% to electricity, much of which is used for cooling. Thus, as the bivalve production moves into autumn, cooling requirements are reduced and production costs fall by as much as 3% relative to a batch available in July.

Implications of Different Survival Rates

The accuracy of hatchery cost estimates derived by this analysis depends on the number of animals surviving at any stage. To test the sensitivity of these results, the survival parameter was changed by plus and minus one and two standard derivatives from the least squares estimate of equation (1). A decrease in the absolute value of the parameter, as in runs 4 and 6, implies a lower mortality rate, whereas an increase in its absolute value leads to higher mortality (runs 5 and 7).

The results of these sensitivity tests are summarized in Table 9. The change in

Table 9. Summary of Simulations to Test Sensitivity of Survival Parameter

Period[a]	Number of Bivalves Surviving, Run (Proportion of Run 1)				Total Feed Cost, Run (Proportion of Run 1)				Cumulative Feed Cost Per Bivalve, Run (Proportion of Run 1)			
	4[b]	5	6	7	4	5	6	7	4	5	6	7
6	1.06	0.95	1.13	0.89	1.05	0.96	1.11	0.91	0.99	1.01	0.98	1.02
10	1.09	0.92	1.21	0.85	1.08	0.93	1.18	0.87	0.99	1.01	0.97	1.02
14	1.11	0.90	1.25	0.82	1.10	0.91	1.23	0.84	0.99	1.01	0.98	1.02
18	1.12	0.89	1.29	0.80	1.11	0.90	1.26	0.82	0.99	1.01	0.98	1.02
22	1.13	0.89	1.32	0.78	1.12	0.90	1.29	0.80	0.99	1.01	0.98	1.02
26	1.14	0.88	1.34	0.77	1.13	0.89	1.31	0.79	0.99	1.01	0.98	1.02
30	1.15	0.87	1.36	0.76	1.13	0.88	1.33	0.78	0.99	1.01	0.98	1.02
34	1.15	0.87	1.38	0.75	1.13	0.88	1.34	0.77	0.99	1.01	0.98	1.02
38	1.16	0.86	1.39	0.75	1.14	0.87	1.36	0.76	0.99	1.01	0.98	1.02
42	1.16	0.86	1.40	0.74	1.15	0.87	1.37	0.76	0.99	1.01	0.98	1.02
46	1.17	0.86	1.42	0.73	1.15	0.87	1.38	0.75	0.99	1.01	0.97	1.02
50	1.17	0.85	1.43	0.73	1.15	0.86	1.39	0.75	0.99	1.01	0.97	1.02
54	1.18	0.85	1.44	0.72	1.16	0.86	1.40	0.74	0.99	1.01	0.97	1.02
59	1.18	0.85	1.45	0.72	1.16	0.86	1.41	0.74	0.99	1.01	0.97	1.03

[a]See Table 7 for explanation of periods.

[b]See Table 6 for assumptions about runs.

the number of surviving bivalves is predictable directly from the parameter changes. Because of the form of the survival function, mortality is assumed to occur at a compound rate. Thus, for increases in mortality, the proportion of bivalves surviving, relative to changes in the base run, falls as the batch moves to later stages. Total feed costs (relative to the base run) fall as well, but not quite so fast. Exactly the opposite is true when mortality rates are reduced.

Perhaps the most important result from this analysis is that despite significant differences in the number of surviving animals (from 0.72 to 1.5 times the base run), cumulative costs per bivalve are never any more than plus or minus 3% of the base run costs. The important implication of this result for hatchery owners is that relatively accurate estimates of per-unit variable costs can be made somewhat independently of good estimates of survival. This is critical in comparing per-unit costs to the potential price of seed to determine when to sell.

Sensitivity of the Relationship of Age and Feed to Size

Although the average cost per bivalve is relatively insensitive to the assumptions made about survival rates, the results of the initial simulations suggest that this is not the case for the relationships between age and size and between feed and size. The importance of these relationships is perhaps best understood by systematically changing the parameters of equations (4) and (10a), assuming the other assumptions in run 1 are constant. The results of this experimentation are summarized in Table 10.

To begin the analysis, the coefficient on the linear term was changed by plus and minus one and two standard deviations. It was readily apparent that although the change

Table 10. Sensitivity of Simulation Results to the Age-Size and Feed-Size Relationships

Period[a]	Average Size of Bivalve					Cumulative Feed Cost per Bivalve[c]				
	Run[b] 8	Run 9	Run 10	Run 11	Run 12	Run 8	Run 9	Run 10	Run 11	Run 12
	(Proportion of Run 1)					(Proportion of Run 1)				
6	0.91	0.91	1.00	1.00	0.91	0.76	0.75	0.58	0.40	0.33
10	0.91	0.90	1.00	1.00	0.90	0.75	0.74	0.38	0.20	0.17
14	0.91	0.90	1.00	1.00	0.90	0.75	0.73	0.29	0.13	0.10
18	0.91	0.89	1.00	1.00	0.89	0.75	0.72	0.25	0.09	0.07
22	0.90	0.88	1.00	1.00	0.88	0.75	0.71	0.22	0.08	0.06
26	0.90	0.88	1.00	1.00	0.88	0.74	0.70	0.20	0.06	0.05
30	0.90	0.87	1.00	1.00	0.87	0.74	0.69	0.18	0.06	0.04
34	0.90	0.86	1.00	1.00	0.86	0.74	0.67	0.17	0.05	0.04
38	0.90	0.86	1.00	1.00	0.86	0.74	0.66	0.16	0.05	0.03
42	0.90	0.85	1.00	1.00	0.85	0.73	0.65	0.15	0.04	0.03
46	0.90	0.84	1.00	1.00	0.84	0.73	0.64	0.15	0.04	0.03
50	0.89	0.83	1.00	1.00	0.83	0.73	0.63	0.14	0.04	0.03
54	0.89	0.83	1.00	1.00	0.83	0.72	0.61	0.14	0.03	0.02
59	0.89	0.82	1.00	1.00	0.82	0.72	0.60	0.13	0.03	0.02

[a]See Table 7 for explanation of periods.

[b]See Table 6 for assumptions about runs.

[c]Total feed costs as a proportion of those in run 1 are the same as the feed costs per bivalve.

in feed costs was more dramatic than the change in bivalve size, both remained about proportional to the change in the parameter, α. Thus, only one of these runs, where α is reduced by two standard deviations, is reported (run 8). This change reduced the estimated size by about 90% of the base run size while reducing feed costs to between 72 to 76% of base run costs.

The next step in the analysis was to change the coefficient on the t^2 term in equation (4). In run 9, the coefficient on the linear term was the same as in run 11, but the absolute value of the coefficient on the squared term was increased by two standard deviations. The effect of this change on both size and feed costs was minimal in the early stages that would be most relevant to a hatchery producing seed to be sold or transferred to growout facilities. At later stages, the effect of this parameter change becomes increasingly important.

The most significant results from Table 10, however, seem to be the large reductions in feed costs due to an increase in the production elasticity of algae in equation (10a). By raising the production elasticity to one standard deviation above its original level, costs fall to just under 60% of the base run (run 10) in period 6 and to 13% by the end of the simulation. Increasing the production elasticity by an additional one standard deviation leads to further, but not quite so dramatic, cost reductions (run 11).

Preliminary Observations about Total Costs of Production

Because data were unavailable from the hatchery on other aspects of the production of oyster seed, it was impossible to implement all parts of the simulation model and estimate costs other than feed costs. However, at least two previous studies contain estimates of the proportion of total costs accounted for by algae feed. On the basis of these estimates, it is possible to make some preliminary observations about total production costs and thus about when during the batch costs will exceed the price that a hatchery operator might receive for the seed.

As reported in Table 4, Im *et al.* (1976) estimate that algae feed costs are about one-third of total production costs; Bolton (1982) reports estimates ranging from 15 to 85% of total costs. Feed costs, as a percentage of total costs, probably increase with age. However, the nature of this relationship is not known; so to obtain some idea of what this wide range in estimates implies for the simulation results above, two kinds of sensitivity analysis were conducted. The first was to assume that feed costs are a constant fraction of total costs (first three panels of Table 11). By assuming feed costs are 33, 59 and 85% of total costs, one is able to reflect all but the low range of Bolton's estimates. It was believed that this low end of the range would apply only to very young bivalves. To reflect increasing relative feed costs, the last two panels in the table assume that costs rise linearly from 15 to 85% of feed costs from period 1 through 59 and period 1 through 30, respectively.

These sensitivity results have important implications for marketing bivalve seed in that Matthiessen and Smith (1979) argued that oyster seed (2 to 3 mm in size) could be purchased at from $2.00 to $5.00 per thousand. Assuming these prices, base run conditions and feed costs at 33% of total, seed would have to be sold prior to the 11th period (5.5 mm shell height) to cover costs if the price were $2.00 per thousand, and before the 13th period (6.9 mm shell height) if the price were to be $5.00 per thousand. This "window of opportunity" would occur much earlier (periods 5 and 6) under the most costly conditions (run 3), but only slightly later (periods 14 and 17) for the least cost scenario (run 10).[15]

These results are remarkably consistent, given the wide range of parameters in the age-size and size-feed relationship reflected in these runs. The "marketing win-

---

15 Since this is a "batch process" and per bivalve feed costs are relatively insensitive to survival rates, it's unlikely that these marketing opportunities would change much if different survival rates were assumed. This does not mean, however, that survival rates are unimportant in terms of the overall profitability of the batch.

Table 11. Alternative Estimates of Total Cumulative Costs of Production ($ Per Oyster)

Period[a]	Run 1[b]	Run 2	Run 3	Run 8	Run 9	Run 10
33.3%[c]						
6	d	d	0.009	d	d	d
10	0.001	0.001	0.061	0.001	0.001	0.001
14	0.008	0.008	0.275	0.006	0.006	0.002
18	0.026	0.026	0.808	0.020	0.019	0.006
22	0.065	0.066	1.845	0.049	0.046	0.014
26	0.135	0.140	3.584	0.100	0.094	0.027
30	0.247	0.262	6.247	0.183	0.169	0.045
(0.002)[e]	11	11	5	12	12	14
(0.005)	13	13	6	14	14	17
(0.010)	15	15	7	16	16	21
59%[c]						
6	d	d	0.005	d	d	d
10	0.001	0.001	0.034	0.001	0.001	c
14	0.004	0.004	0.155	0.003	0.003	0.001
18	0.015	0.015	0.456	0.011	0.011	0.004
22	0.037	0.037	1.041	0.027	0.026	0.008
26	0.076	0.079	2.022	0.057	0.053	0.015
30	0.139	0.148	3.526	0.103	0.095	0.025
(0.002)[e]	12	13	6	13	13	16
(0.005)	15	15	6	16	16	20
(0.010)	17	17	8	18	18	24
85%[c]						
6	d	d	0.004	d	d	d
10	0.001	0.001	0.024	d	d	d
14	0.003	0.003	0.108	0.002	0.002	0.001
18	0.010	0.010	0.317	0.008	0.007	0.003
22	0.026	0.026	0.722	0.019	0.018	0.006
26	0.045	0.055	1.404	0.039	0.037	0.010
30	0.097	0.103	2.448	0.072	0.067	0.018
(0.002)[e]	13	13	6	14	14	18
(0.005)	16	16	7	17	17	22
(0.010)	18	18	9	20	20	26
15-85%						
6	d	d	0.015	d	d	d
10	0.002	0.002	0.078	0.001	0.001	0.001
14	0.008	0.008	0.298	0.006	0.006	0.002
18	0.024	0.024	0.758	0.018	0.018	0.006
22	0.054	0.054	1.522	0.040	0.038	0.012
26	0.100	0.103	2.642	0.074	0.070	0.020
30	0.164	0.174	4.161	0.122	0.112	0.030
(0.002)[e]	11	11	5	11	11	14
(0.005)	13	13	5	14	14	18
(0.010)	15	15	6	16	16	21
15-85%[c]						
6	d	d	0.011	d	d	d
10	0.001	0.001	0.055	0.001	0.001	d
14	0.006	0.005	0.197	0.004	0.004	0.002
18	0.016	0.015	0.480	0.012	0.011	0.004
22	0.033	0.033	0.935	0.025	0.023	0.007
26	0.060	0.062	1.584	0.044	0.041	0.012
30	0.100	0.103	2.448	0.072	0.066	0.018
(0.002)[e]	12	12	5	12	12	15
(0.005)	14	14	6	15	15	20
(0.010)	17	17	6	18	18	25

[a]See Table 7 for explanation of periods.

[b]See Table 6 for assumptions about runs.

[c]These are the percentages that feed costs are assumed to be of total cost. In the last two panels, costs are assumed to increase linearly from 15 to 85% of feed cost from periods 1 through 30 and periods 1 through 59, respectively.

[d]Less than 0.0005.

[e]Actual period during which costs reach $0.002, $0.005 and $0.010, respectively, per bivalve.

dow" is extended by only 2 or 3 weeks if feed costs actually turn out to be 85% rather than 33% of total production costs or if feed costs are assumed to rise to 85% of total costs over 30 rather than 59 periods. A doubling of the price (from $0.005 to $0.01 would extend the marketing period by only an additional 2 to 3 weeks as well.[16]

It is also important to recognize that during these "break-even" marketing periods, it is estimated that the size of the seed is much larger than the 2 to 3 mm on which the prices are based. This implies that the hatchery owners may be in an advantageous bargaining position to sell seed that meets the size requirements earlier, thus reducing costs, or by asking for a higher price for larger seed, with its higher survival rates during growout.

VI - SUMMARY AND CONCLUSIONS

The bivalve aquacultural industry in the northeastern United States has been strengthened in recent years by declining and fluctuating natural harvests in the surrounding coastal region. The hatchery industry, in which the bivalves are grown only to a small seed size, has gained acceptance as an important link in the aquacultural process; the firms in the industry are potentially able to provide a consistent, reliable source of seed with which the bivalve aquaculturist begins the production process. They have also gained acceptance as a seed supplier to local governments, who artificially seed depleted natural bivalve beds with the seed purchased from hatcheries. Hatcheries as an industry are likely to play a larger role in the overall production of bivalves if they can overcome some of the technical and economic problems with which they must contend.

The techniques involved in the production of bivalve seed in hatcheries have evolved only recently, yet are technically sophisticated compared to other open-water aquacultural techniques. The production processes differ, but each requires a precise sequence of steps under carefully-controlled and -monitored environmental conditions. Mature adult clams or oysters taken from hatchery-maintained broodstock are induced to spawn. The gametes are transferred into large, conical larvae tanks where they freely circulate in warm, algae-rich seawater. The concentrated algae used for feed are grown in the hatchery or taken from seawater. After several weeks, the bivalves undergo a metamorphosis where they change from free-swimming larvae to bottom-dwelling spat. Prior to this time, the animals are transferred to set tanks where they are induced to attach to culch -- a material purchased or produced in the hatchery generally from crushed bivalve shell. Once the animals have passed through the delicate setting period, they are transferred to the juvenile tanks where they can be exposed to large quantities of algae and seawater. The animals remain in these tanks until they reach a suitable size for transfer to open-waters.

There are a number of production options available to the hatcherist, the most apparent being the design of the production system and the source of inputs. The algae production method is the greatest cause of variations affecting cost among hatcheries. These methods include the centrifuging of algae cells from seawater, continuous culture methods, or a batch process in which production is separated into distinct stages. The type of culch used and the number of animals produced as culched or culchless (attached or non-attached to the culch) also differ among hatcheries. The hatcherist has a great deal of discretion regarding the optimum survival rate of the animals in relation to the quantity and quality of inputs used. Finally, and most central to this research, is the choice of when to sell seed.

Despite the growth of the industry and advancements in the technical processes of hatcheries, little is known about the economics of the hatchery. There have been several important studies on this aspect of hatcheries (Im, *et al.*, 1976 and by Im and Langmo, 1977). These and other studies fail to pay sufficient attention to the wide variation that exists among hatcheries and to break

---

[16] This price was included because there has undoubtedly been an increase in the price of seed between that reported by Matthiessen and Smith (1979) and 1982, the year for which the algae cost data apply.

cost down by period of the animal's life. Estimates of cost by period are essential to determine the optimal time (*viz.* size) at which to sell the seed, inasmuch as the optimal selling age is at the point where marginal cost is equal to the price that could potentially be received.

The purpose of this research is to contribute to the existing body of knowledge about the hatchery industry by examining the factors that influence production costs. The research is partly based on observations of a working hatchery located in the northeast and on current literature, with particular attention paid to the variations that exist among hatcheries. The data collected were used to develop the framework for a computer program that could estimate variable production costs. The program routine was intended to enable the individual hatcherist to determine the optimal size at which to sell the bivalve seed.

Creation of this program required the development of an algorithm that simulates the production process. The model sums costs at different stages of animal growth, which are further broken down into smaller units of time. In this way a cost can be computed at any given point in the animals' lives. The cost incurred during any given period of time is a function of the quantity of inputs used and their prices. Thus, the major component of the model was designed to determine the quantity of inputs used and the cost incurred during any given period.

There was sufficient information available from the literature and hatchery observations to simulate the overall production system. Unfortunately, records concerning the expenditures did not distinguish between hatchery and open-water field operations nor did they break costs down by stage. Consequently, data were not available to allow for all aspects of the model to be completed in detail. Data on the batch algae production process used by the hatchery were, however, more readily isolated from the rest of the hatchery because of the physical separation of the two systems in the hatchery. Thus, the focus of the empirical analysis was on the estimation of costs of algae production and is justified since feeding costs are recognized as a large proportion of total costs.

The FORTRAN computer program that is based on the model focuses on the cost of producing algae feed, but as more details become available about other hatchery production processes it would be a simple matter to incorporate them into the existing FORTRAN code. The computer program calculates the cost, or internal price of algae to the hatchery, and then combines this information with survival and growth rates and feeding efficiencies to determine total (and per unit) feeding costs to raise a batch of bivalves to various sizes. The algorithms are used to examine the sensitivity of the algae production parameters and to the parameters of the bivalve growth, survival and feeding relationships.

The model is applied to raising Atlantic Oyster seed, using data on algae production from the hatchery and data on survival rates, feeding efficiency and growth rates available from the literature. Twelve initial simulations were made. The output from each contained a summary of input prices and requirements and costs for bivalves at different periods throughout the production process, along with the age, size and number of bivalves and the total and average costs per bivalve.

The 12 simulations reflect a range of assumptions about survival rates, growth rates and algae feeding rates. In the base run, feed requirements are related directly to animal size. The feed cost per surviving bivalve remains less than one cent up to 100 days but rises rapidly thereafter. In a second run, based on the same data to relate feed directly to age, feed cost remains lower for the first 100 days, but then rises to 1.3 times the base run.

As might be expected, the feed costs per bivalve are extremely sensitive to the parameters of the feeding relationships and to the age-size relationship. This is not particularly encouraging because there seem to be few sources of data in the literature from which to estimate these relationships. The one encouraging thing is that feed costs are less sensitive to these parameters for very young bivalves than for older ones, and it is the young bivalve that is of most interest to a hatchery. Another important result is that for wide differences in the numbers of animals surviving, the feed cost per surviving animal (including the accumulated cost of feeding those that did not survive) varied by no more than plus or minus 3%. Having good estimates of feed costs per animal in the face of considerable uncertainty about survival rates may well facilitate planning when to sell the bivalve seed or transfer them for growout.

To explore the implications of these feed cost simulations for the marketing of bivalve seed, some rough estimates of total costs of production per bivalve were developed on the basis of others' estimates of the proportion that feed costs are of total costs. Total costs were estimated for several simulations assuming feed costs were 1) one-third, 2) nearly 60%, 3) 85% of total costs, and 4) a linear function of total costs starting at 15% and rising to 85% by the end of the production period. While the time period at which price just covered costs differed significantly over the scenarios, within a given scenario the "break-even" marketing periods varied only by a week or two in spite of diverse assumptions about the fraction feed costs are of total costs.

In conclusion, this research demonstrates that it is possible and practical to estimate algae and feeding costs for a seed producing bivalve hatchery using a computer program to simulate the production system. The most optimistic assumptions suggest that bivalve seed can be produced and marketed profitably. These conclusions must be qualified by the fact that sufficient data were not available from the hatchery under study to estimate all components of total cost. Collection of these types of data would help in further testing of the algorithm as well as contributing to an overall understanding of hatchery costs at each stage of production. Equally important from a research perspective is the need for better data on hatchery survival rates and the size-age and feeding rates. What data do exist in the literature on which to estimate these rates lead to quite different implications for feed costs, and none of the existing data sets contains more than a few observations. Thus, in the absence of more extensive research data on which to estimate these important physical relationships, anyone attempting to use this kind of software for management decisions is well advised to use information that is specific to a particular hatchery.

References

Agnello, R. J. and L. P. Donnelley. "Externalities and Property Rights in the Fisheries". Land Economics, 52 (1976):518-529.

Baab, J. S., G. L. Hamm, K. C. Hanes, A. Chu and O. A. Roels. "Shellfish Mariculture in an Artificial Upwelling System." Proceedings of the National Shellfisheries Association, 63 (1973):63-68.

Bardach, J. E., J. Ryther and W. McLarney. Aquaculture - The Farming and Husbandry of Freshwater and Marine Organisms. New York: John Wiley and Sons Inc., 1972.

Bolton, E. (ed.) Intensive Marine Bivalve Cultivation in a Controlled Recirculating Seawater Prototype System. DEL-SG-07-82, University of Delaware, 1982.

Breese, W. P. and R. E. Malouf. "Hatchery Manual for the Pacific Oyster". Oregon State University Sea Grant College Program, Agriculture Exper. Station, Special Report Number 443, December, 1975.

Castagna, M. and J. N. Kraeuter. "Mercenaria Culture Using Stone Aggregate for Predator Protection". Proceedings of the National Shellfisheries Association, 67 (1977):1-6.

Claus, G. and C. Adler. "Optimal Conditions for Oysters Grown in Closed Environments". Proceedings of the Marine Technical Society, 1 (1970):671-679.

Donohue, M. J., D. A. Storey and P. G. Allen. "Technical and Economic Feasibility of Alternative Aquacultural Production Systems for American Oysters and Other Bivalve Mollusks in New England". Massachusetts Agricultural Experiment Station Research Bulletin, Number 668, June 1981.

Dressel, D. M. and D. S. Fitzgibbon. "The U.S. Molluscan Shellfish Industry". United States Dept. of Commerce, National Oceanic and Atmospheric Administration, National Marine Fisheries Service, Washington, D.C., pp. 255-261, in "Drugs and Food from the Sea; Myth or Reality?", P.N. Kaul, and C.J. Sindermann (eds.), University of Oklahoma, 1978.

Eldridge, P., A. Eversole and J. Whetstone. "Comparative Survival and Growth of Hard Clams Mercenaria Mercenaria, Planted in Trays Subtidally and Intertidally at Varying Densities in a South Carolina Estuary". Proceedings of the National Shellfisheries Association, 69 (1979):30-39.

Emmel, T. G. Population Biology. New York: Harper and Row, Publishers, 1976.

Epifanio, C. E. "Culture of Bivalve Mollusks in Recirculating Systems: Nutritional Requirements". Proceedings of the First International Conference on Aquaculture Nutrition, , pp. 173-194, October, 1975.

Epifanio, C. E., C. M. Logan and C. Turk. "Culture of Six Species of Bivalves in a Recirculating Seawater System". Proceedings of the 10th European Symposium of Marine Biology, 1976.

Freedman, M. and T. Morris. "A Dwindling Harvest Industry". Sunday Newsday, Long Island, New York, Suffolk Edition, p.3, March 6, 1983.

Galtsoff, P. S. "The American Oyster Crassostrea Virginica Gmelin". Fishery Bulletin of the Fish and Wildlife Service, 64(1964):397-301.

Gates, J. M., G. C. Matthiessen and C. A. Griscon. "Aquaculture in New England". Sea Grant University of Rhode Island, Marine Technical Report Series Number 18, Kinston, 1974.

Goldstein, M. "Plan for Clams - Without Rocking the Boat". Sunday Newsday, Long Island Newspaper, Suffolk Edition, p.21, March 7, 1983

Hanson, J.A. (ed.) "Open Sea Mariculture". Pennsylvania: Hutchinson and Ross, 1974.

Hardin, G. "The Tragedy of the Commons". Science, 162 (1968):1243-1247.

Henderson, S. P. "Shellfish Hatcheries, an Industry View," in Drug and Food from the Sea: Myth or Reality? P.N. Kaul and C.J. Sindermann, eds., University of Oklahoma, 1978.

Im, K. H., R. S. Johnston and R. D. Langmo. "The Economics of Hatchery Production of Pacific Oyster Seed: A Research Progress Report". Proceedings of the National Shellfisheries Association, 66 (1976):81-94.

Im, K. H. and R. D. Langmo. "Economic Analysis of Producing Pacific Oyster Seed in Hatcheries". Proceedings of the National Shellfisheries Association, 67 (1977):17-28.

Kochiss, J. M. "Oystering from New York to Boston". Connecticut: Wesleyan University Press, 1974.

Korringa, P. "Farming the Cupped Oyster of the Genus Crassostrea". New York: Elsevier Scientific Publishing Co., 1976.

Loosanoff, V. L. and H. C. Davis. "Rearing of Bivalve Mollusks". Advances in Marine Biology 1(1963a):1-136.

Loosanoff, V. L. and H. C. Davis. "Shellfish Hatcheries and Their Future". United States Department of the Interior, Fish and Wildlife Service, Washington, D.C., Commercial Fisheries Review, 25, 1963b.

Mansfield, E. "Microeconomics - Theory and Application". New York: WW Norton and Co., 1979.

Manzi, J. J., V. G. Burrell Jr. and H. Q. Clawson. "Commercialization of the Hard Clam (Mercenaria mercenaria), Mariculture in South Carolina". Preliminary Report , in J. W. Avault Jr. (ed.), Proceedings of the Twelfth Annual Meeting World Mariculture Society. Louisiana State University, 1982a.

Manzi, J. J., V. G. Burrell Jr., H. Q. Clawson, F. S. Stevens and M. B. Maddox. "Hard Clam, Mercenaria mercenaria (Linne), Mariculture in South Carolina", draft for review only, Oct., 1982b.

Matthiessen, G. C. "Review of Oyster Culture and the Oyster Industry in North America". Number 2528, Woods Hole Oceanographic Institute, 1970.

Matthiessen, G. C. "The Oyster Industry of Massachusetts and the Introduction of Exotic Species" in R. Mann (ed.), Exotic Species and Mariculture, pp. 213-224, MIT Press, Cambridge, 1979.

Matthiessen, G. C. and L. J. Smith. "Analysis of Methods for the Culture of Crassostrea virginica in New England". Proceedings of the World Mariculture Society, 10 (1979):609-623.

Matthiessen, G. C. and R. C. Toner. "Possible Methods of Improving the Shellfish Industry of Martha's Vineyard, Duke's County, Massachusetts". Marine Research Foundation. Inc., Edgartown, Massachusetts, 1966.

McHugh, J. and A. Williams. "Historical Statistics of the Fisheries of the New York Bight", New York Sea Grant Institute, Albany, New York, 1976.

Miller, R. "A View Through the Porthole by a Commercial Fisherman" in O. Terry (ed.) Proceedings of the Symposium on Aquaculture in New York State, 1977.

Myers, J. M. Production and Feeding Costs of a Bivalve Seed Hatchery. Unpublished M.S. thesis, June 1985.

Pillay, T. V. R. "The State of Aquaculture 1975". FAO Technical Conference on Aquaculture, May 26 - June 2, 1976, Kyoto, Japan, pp. 1-13.

Regional Marine Resources Council, Committee of the Nassau-Suffolk Regional Planning Board. "Guidelines for Management of Long Island Hard Clam Resources", pp. 2-13, 1974.

Rhodes, E. W. and W. S. Landers. "Growth of Oyster Larvae Crassostrea virginica of Various Sizes in Different Concentrations of the chrysophyte, Isochrysis galbana". Proceedings of the National Shellfisheries Association, 63(1973):53-59.

Rhodes, T. "Comments of a Bayman" in O. Terry and D. Chase (eds.) Proceedings of the Symposium on Mariculture in New York State, Southhampton College, New York, 1974.

Ritchie, T. P. "A Comprehensive Review of the Commercial Clam Industries in the United States". Delaware Sea Grant Program Publication, #DEL-SG-26-76, United States Government Printing Office, Washington, D.C., 1977.

Smith, B. STOP Project Director, interviews, 1982-83.

Terry, O. W. "Aquaculture," MESA New York Bight Atlas Monograph Number 17, New York Sea Grant Institute, Albany, New York, June 1977.

Thomas, G. B., Jr. Calculus and Analytic Geometry. 3rd ed. Reading, Mass.: Addison-Wesley Publishing Co., Inc. 1960.

Ukeles, R. "View on Bivalve Larvae Nutrition". Proceedings of the First International Conference on Aquaculture Nutrition, pp. 127-162, 1975.

United States Department of Commerce, NOAA, NMFS. "New York Landings, Annual Summary 1973". Current Fisheries Statistics Number 6412, p.2, Washington, D.C., 1974.

United States Department of Commerce, NOAA, NMFS. "Fishery Statistics of the United States". Washington, D.C., 1979.

APPENDIX A

```
      PROGRAM ORGNSM
C     --------------

C  ORGNSM:
C  -------
C  VARIABLES DO NOT CHANGE WHEN ORGANISM CHANGES

C  STRUCTURE:  GENERAL PROGRAM INFORMATION
C              OPTION TO CALL SUBR. INFORM TO DISPLAY PROGRAM INFOR.
C              CALLS SUBR. READIN TO READ IN ALL DATA
C              CALLS SUBR. BODY TO DO COMPUTATIONS
C              OFFERS OPTION TO RERUN PROGRAM WITH MINOR CHANGES


C??????????????????????????????????????????????????????????????????????
C  GENERAL PROGRAM INFORMATION:
C  ---------------------------

C  TITLE :  ORGANISM COST SIMULATION MODEL
C  AUTHOR:  JULIA A. MYERS
C  PLACE :  CORNELL UNIVERSITY
C  DATE  :  DECEMBER 6, 1986

C  LANGUAGE: FORTRAN FF  - IBM COMPATIBLE
C            NONSTANDARD - MIXED CHARACTER COMMON BLOCKS, INCLUDE,
C                          '$' IN VARIABLE NAMES
C  DOCUMENTATION: USER SPREAD-SHEET (FOR INPUTTING DATA)

C     THIS INTERACTIVE FORTRAN PROGRAM CONTAINS THE BASIC STRUCTURE
C  OF A MODEL NECESSARY TO SIMULATE THE COST OF PRODUCING
C  ANY ORGANISM THAT GROWS BY STAGES. THE COST AND QUANTITY OF INPUTS
C  IS DETERMINED AT EACH STAGE OF THE ORGANISMS' LIVES AND AT
C  DIFFERENT PERIODS IN THOSE STAGES.
C     THE PROGRAM'S MODULAR DESIGN ALLOWS SUBROUTINES TO BE ADDED
C  FOR SPECIFIC INPUTS AND STAGES OF THE PRODUCTION PROCESS
C  THEREFORE, IT IS POSSIBLE TO DETERMINE ALL OR ONLY PART OF THE TOTAL
C  COST OF PRODUCTION.  EACH SUBROUTINE CONTAINS A DESCRIPTION OF
C  THE PROGRAM VARIABLES THAT MUST BE RETURNED TO THE PROGRAM.
C     THE CURRENT CAPABILITIES OF THE PROGRAM ARE INITIALIZED IN BLOCK
C  DATA, BLK1.  THESE MUST BE RESET WHEN SUBROUTINES ARE ADDED OR
C  DELETED.  SEE THE ACCOMPANYING PROGRAM DOCUMENTATION.
C     THE PROGRAM WAS TESTED FOR THE PRODUCTION OF BIVALVE SEED IN AN
C  ENCLOSED HATCHERY.  THUS, SOME SUBROUTINES ARE SPECIFIC TO
C  THIS ORGANISM AND THE APPROPRIATE CHANGES MUST BE MADE FOR OTHERS.


C  NAME    DESCRIPTION OF REQUIRED ASSOCIATED FILES INCLUDED IN PROGRAM
C  ------  ------------------------------------------------------------
C  IPAR1   PARAMETERS NECESSARY FOR GENERAL COMMON BLOCK VARIABLES
C  ICMAIN  VARIABLES IN COMMON NECESSARY TO RUN GENERAL PROGRAM
C  ICUNIT  VAR. IN COMMON FOR UNIT NUMBERS OF READ AND WRITE
C  ICBLK1  VAR. IN COMMON USED IN SUBR. BLOCK DATA BLK1
C  ICBLKO  VAR. IN COMMON, USED IN SUBR. BLOCK DATA BLK1 FOR ORG.
C  ICORG   VAR. IN COMMON SPECIFIC TO ORGANISM
C  (NOTE:  OTHER FILES MAY BE REQUIRED FOR ORGANISM SPECIFIC ROUTINES)
C----------------------------------------------------------------------


C----------------------------------------------------------------------
C FLOW OF CONTROL:
C                                  ORGNSM
C                                    I
C    ----------------------------------------------------
C     I          I                                     I
C   INFORM     READIN                                 BODY
C                I                                     I
C                I               ---------------------------------
C                I               I    I    I          I     I     I
C                I               I    I    I          I     I     I
C                I           NUMBRB SIZE QUANT1-6 COSTJ  WRTOUT WOUTN1-6
C                I
C                I
C       --------------------------------------------------
C       I          I          I           I    I    I    I
C       I          I          I           I    I    I    I
C     READS      READO     READN1-6    ROUT ROUTS ROUTO ROUTN1-6
C       I          I          I
C       I          I          I
C   ---------    ROUTO     ROUTN1-6
C    I       I
C    I       I
C ROUTS PRICE1-6
C            I
C      ------------
C      I          I
C   RIN$1-6    ROUT$1-6
C----------------------------------------------------------------------


C  NAME    DESCRIPTION OF REQUIRED SUBROUTINES AND FUNCTIONS BY APPEARAN
C  ------  ------------------------------------------------------------
C  ORGNSM  MAIN PROGRAM
C  INFORM  READS OUT INFORMATION CONCERNING THE PROGRAM
C  READIN  READS IN OPTION INFO AND CALLS ALL SUBROUTINES TO READ IN
C  ROUT    READS OUT DATA FROM READIN
C  READS   READS IN DATA TO RUN GENERAL PROGRAM
C  ROUTS   READS OUT DATA FROM READS
C  READO   READS IN DATA SPECIFIC TO ORGANISM
C  ROUTO   READS OUT DATA FROM READO
C  READN1  READS IN DATA TO COMPUTE QUANT OF INPUT 1
C  READN2  READS IN DATA TO COMPUTE QUANT OF INPUT 2
C  READN3  READS IN DATA TO COMPUTE QUANT OF INPUT 3
C  READN4  READS IN DATA TO COMPUTE QUANT OF INPUT 4
C  READN5  READS IN DATA TO COMPUTE QUANT OF INPUT 5
C  READN6  READS IN DATA TO COMPUTE QUANT OF INPUT 6
C  ROUTN1  READS OUT DATA ENTERED IN READN1
C  ROUTN2  READS OUT DATA ENTERED IN READN2
C  ROUTN3  READS OUT DATA ENTERED IN READN3
C  ROUTN4  READS OUT DATA ENTERED IN READN4
C  ROUTN5  READS OUT DATA ENTERED IN READN5
C  ROUTN6  READS OUT DATA ENTERED IN READN6
C  PRICE1  COMPUTES PRICE OF INPUT 1
C  PRICE2  COMPUTES PRICE OF INPUT 2
C  PRICE3  COMPUTES PRICE OF INPUT 3
C  PRICE4  COMPUTES PRICE OF INPUT 4
C  PRICE5  COMPUTES PRICE OF INPUT 5
C  PRICE6  COMPUTES PRICE OF INPUT 6
C  BODY    LOOPS THRU EACH MONTH TO CALL $ AND QUANT SUBR.
C  NUMBRB  COMPUTES NUMBER OF ORGANISM AT PERIOD,J
C  SIZE    COMPUTES SIZE OF ORGANISM AT PERIOD J
C  QUANT1  COMPUTES QUANT OF INPUT 1 USED IN J
C  QUANT2  COMPUTES QUANT OF INPUT 2 USED IN J
```

```
C  QUANT3   COMPUTES QUANT OF INPUT 3 USED IN J
C  QUANT4   COMPUTES QUANT OF INPUT 4 USED IN J
C  QUANT5   COMPUTES QUANT OF INPUT 5 USED IN J
C  QUANT6   COMPUTES QUANT OF INPUT 6 USED IN J
C  COSTJ    SUMS AND COMPUTES COST OF INPUTS IN PERIOD, J
C  WRTOUT   PRINTS COSTS AND QUANTITIES
C  WOUTN1   PRINTS ADDITIONAL INFO GENERATED ABOUT INPUT 1
C  WOUTN2   PRINTS ADDITIONAL INFO GENERATED ABOUT INPUT 2
C  WOUTN3   PRINTS ADDITIONAL INFO GENERATED ABOUT INPUT 3
C  WOUTN4   PRINTS ADDITIONAL INFO GENERATED ABOUT INPUT 4
C  WOUTN5   PRINTS ADDITIONAL INFO GENERATED ABOUT INPUT 5
C  WOUTN6   PRINTS ADDITIONAL INFO GENERATED ABOUT INPUT 6
C  AGENO    FUNCTION RELATING AGE TO SURVIVAL
C  AGESZE   FUNCTION RELATING AGE TO SIZE
C  BLK1     BLOCK DATA FOR GENERAL PROGRAM
C  BLKO     BLOCK DATA FOR ORGANISM DEPENDENT VARIABLES
C  (NOTE: OPTIONAL SUBROUTINES LISTED UNDER APPROPRIATE SUBROUTINES)
C  ------------------------------------------------------------------


C  NAME   TYPE COMMON DIMENS DESCRIPTION OF COMMON BLOCK VARIABLES
C  ------ ---- ------ ------ ---------------------------------------
C  AGEUPJ R    CMAIN  MAXJI  AGE OF ORGANISM AT END OF PERIOD
C  ANYNOK I    CMAIN  NOOFN  TO DETERMINE IF ANY N ARE TO BE COMPUTED
C  BEGINI I    CMAIN         STAGE TO BEGIN COST CALCULATIONS.
C                            OPTION 1, BEGIN AT STAGE(BEGINI+1)
C  BEGINM I    CMAIN         FIRST MONTH OF PRODUCTION
C  BEGINJ I    CMAIN  NOOFI  NO. OF THE 1ST PERIOD OF EACH STAGE
C  CSOOFJ R    CMAIN  MAXJI  COST/ORGANISM - AVERAGE IN STAGE
C  CSOUPJ R    CMAIN  MAXJI  COST/ORGANISM UP TO END OF PERIOD,J
C  CSNOFJ R    CMAIN  MAXJI  COST OF EACH INPUT,N, DURING PERIOD,J
C                     NOOFN
C  CSNUPJ R    CMAIN  MAXJI  COST OF EACH INPUT,N, UP TO END OF PERIOD,J
C                     NOOFN
C  CSOFJ  R    CMAIN  MAXJI  COST OF ALL INPUTS,N, DURING PERIOD,J
C  CSOP1  R    CMAIN         OPTION 1: COST UP TO 1ST STAGE TO CALCULATE
C  CSUPJ  R    CMAIN  MAXJI  COST OF ALL INPUTS,N, UP TO END OF PERIOD,J
C  DATE   C*50 CMAIN         CURRENT DATE
C  DAYUPJ R    CMAIN  MAXJI  TOTAL ELAPSED DAYS UP TO END OF PERIOD,J
C  DAYOP1 R    CMAIN         OPTION 1: TOTAL ELAPSED DAYS UP TO 1ST STAG
C  DAYSI  R    CMAIN  NOOFI  TOTAL DAYS IN EACH STAGE,I
C  DAYSJ  R    CMAIN  NOOFI  NO OF DAYS IN EVERY PERIOD,J, OF STAGE,I
C  ENDJ   I    CMAIN         LAST PERIOD CALCULATED COST FOR
C  ENDM   I    CMAIN         LAST MONTH OF THE PRODUCTION SEASON
C  IOFJ   I    CMAIN  MAXJI  STAGE THAT EACH PERIOD IS IN
C  JPERI  I    CMAIN  NOOFI  NUMBER OF PERIODS,J, IN STAGE,I
C  MAXI   I    CPAR1         MAXIMUM NO. OF STAGES,I, POSSIBLE:  PAR=6
C  MAXJ   I    CPAR1         MAX NO. OF PERIODS,J, IN STAGES: PAR=10
C  MAXJI  I    CPAR1         MAX NO PERIODS: PAR = MAXJ X MAXN
C  MAXM   I    CPAR1         MAX NO OF MONTHS:  PAR=12
C  MAXN   I    CPAR1         MAX NO OF INPUTS, N:  PAR=6
C  MAXT   I    CPAR1         MAX NO OF TYPES,T IN INPUT N: PAR=6
C  MOFJ   I    CMAIN  MAXJI  MONTH THAT J OCCURS IN
C  M1     I    CMAIN         MONTH TO START CALCULATIONS ON
C  M2     I    CMAIN         MONTH TO END CALCULATIONS ON
C  NAMEF1 C*10 CMAIN         NAME OF INPUT FILE USED
C  NAMEM  C*10 CBLK1  CMAIN  NAME OF EACH MONTH, M
C  NAMES  C*25 CMAIN         NAME OF SYSTEM USED
C  NOOFO  I    CPAR1         NO. OF ORGANISM TYPES, O:  PAR=2
C  NOOFI  I    CPAR1  NOOFI  NO. OF STAGES, I:  PAR=5
C  NOOFN  I    CPAR1         NO OF INPUTS, N:  PAR=6
C  NOOFT  I    CBLK1  NOOFN  NO OF TYPE,T, OF EACH INPUT,N
C  NQBYM  C*1  CMAIN  NOOFN  DOES QUANTITY OF INPUT N VARY BY MONTH?
C  N$BYM  C*1  CMAIN  NOOFN  DOES PRICE OF INPUT,N, VARY BY MONTH?
C  ONOBEG R    CMAIN         NO. OF ORGANISMS AT BEGIN
C  ONOOFJ R    CMAIN  MAXJI  AVG. NO. OF ORGANISMS DURING EACH PERIOD,J
C  ONOUPJ R    CMAIN  MAXJI  NO. OF ORGANISMS AT END OF EACH PERIOD,J
C  OTYPE  I    CMAIN         TYPE OF ORGANISM TO DO CALCULATIONS FOR
C  OPN$OK I    CBLK1  NOOFN  CAN PROGRAM COMPUTE PRICE OF INPUT,N?
C                            0=YES, 1=NO
C  OPNNEW C*1  CMAIN  NOOFI  DOES THE USER WANT TO ENTER N IN I?
C                     NOOFN
C  OPNOK  I    CBLK1  NOOFI  CAN PROGRAM COMPUTE INPUT,N, IN STAGE,I?
C                     NOOFN  0=YES, 1=NO
C  OPTION I    CMAIN         OPTIONS 1-3 THAT IS CHOSEN TO COMPUTE Q
C  OPQOK  I    CBLK1  3      CAN PROGRAM DO EACH OPTION FOR COMPUTING?
C                            0=YES, 1=NO
C  OPTOK  I    CBLK1  NOOFN  CAN USER CHANGE NO. OF TYPES,T OF INPUT,N?
C                            0=YES, 1=NO
C  OUTQN  I    CBLK1  NOOFN  IS IT POSSIBLE TO WRITE ADDITIONAL INFO N?
C  PASEDM I    CMAIN  MAXJI  IF PAST MONTHS OF PROD., SET TO 1
C  PRICET R    CMAIN  NOOFN  PRICE OF TYPE,T, OF INPUT,N, IN MONTH,M.
C                     MAXT
C                     MAXM
C  QOPN$  C*1  CMAIN  NOOFN  DOES USER WANT TO COMPUTE PRICE OF INPUT,N?
C  QOUTQN C*1  CBLK1  NOOFN  PRINT OUT ADDITIONAL INFO ABOUT INPUT N?
C  RF     I    CUNIT         UNIT NUMBER FOR READ FILE
C  RI     I    CUNIT         UNIT NUMBER FOR READ INTERACTIVE
C  SZEOFJ R    CMAIN  MAXJI  AVG SIZE OF ORGANISM DURING PERIOD
C  SZEUPJ R    CMAIN  MAXJI  SIZE OF ORGANISM AT END OF PERIOD
C  TOTALI I    CMAIN         NO. OF STAGES TO DO COMPUTATIONS FOR (OP=1)
C  TQUANI R    CMAIN  NOOFI  QUANTITY OF TYPE,T, OF INPUT,N, USED IN
C                     NOOFN  MONTH,M FOR STAGE,I.
C                     MAXT
C                     MAXM
C  TQUANJ R    CMAIN  MAXJI  QUANTITY OF TYPE,T, OF INPUT,N, USED IN
C                     NOOFN  MONTH,M, IN PERIOD, J.
C                     MAXT
C  WKSUPJ R    CMAIN         ELAPSED WEEKS UP TO END OF PERIOD,J
C  WF     I    CUNIT         UNIT NUMBER FOR WRITE FILE
C  WI     I    CUNIT         UNIT NUMBER FOR WRITE INTERACTIVE
C  WO     I    CUNIT         UNIT NUMBER TO WRITE OUTPUT FILE
C  WON    I    CUNIT  NOOFN  UNIT TO WRITE TO N DIFFERENT OUTPUT FILES
C  (NOTE: OTHER VARIABLES MAY BE LISTED UNDER OPTIONAL SUBROUTINES)
C  ------------------------------------------------------------------


C  NAME   TYPE COMMON DIMENS DESCRIPTION OF COMMONS USED FOR ORGANISMS
C  ------ ---- ------ ------ ---------------------------------------
C  DAYSI1 R    CORG          DAYS IN I=1, CONDITIONING
C  EQNO   C*50 CBLKO  1      EQUATION RELATING AGE TO SURVIVAL
C  EQNOA  R    CBLKO         PARAMETER A FOR AGE NUMBER EQUATION
C  EQNOAF R    CBLKO         PARAMETER A RECOMMENDATION FOR AGE/NO EQ.
C  EQQ5   C*50 CBLKO  2      EQUATION RELATING AGE TO QUANT N=5
C  EQQ5A  R    CBLKO  2      PARAMETER A FOR QUANT N=5 EQUATION
C  EQQ5AF R    CBLKO  2      RECOMMENDATION FOR PARAM. A FOR QUANT5 EQ
C  EQQ5B  R    CBLKO  2      PARAMETER B FOR QUANT N=5 EQUATION
C  EQQ5BF R    CBLKO  2      RECOMMENDED VALUE FOR PARAM B FOR QUANT5 EQ
C  EQSZE  C*50 CBLKO  1      EQUATION RELATING AGE TO SIZE
C  EQSZA  R    CBLKO         PARAMETER A FOR AGE SIZE EQUATION
C  EQSZAF R    CBLKO         RECOMMENDED VALUE FOR PARAM A FOR SIZE EQ
C  EQSZB  R    CBLKO         PARAMATER B FOR AGE SIZE EQUATION
C  EQSZBF R    CBLKO         RECOMMENDED VALUE FOR PARAM B FOR SIZE EQ
C  FILTER R    CORG   MAXI   FEEDING FILTERING RATE, AS A FRACTION
C  NAME   C*1  CBLKO         NAME OF ORGANISM PROGRAM IS SPECIFIC TO
C  NAMEO  C*10 CBLKO  NOOFO  NAME OF ORGANISM TYPES
C  NAMEI  C*10 CBLKO  NOOFI  NAME OF EACH STAGE, I
C  NAMEN  C*10 CBLKO  NOOFN  NAME OF EACH INPUT, N
```

```
C  NAMENU C*5  CBLKO  NOOFN  UNIT NAME OF INPUT, N
C  NAMET  C*10 CBLKO  NOOFN  NAME OF EACH TYPE T, OF INPUT, N
C  NOOI1  R    CORG          NO. ORGANISM IN I=1, CONDITIONING STAGE
C  N5AVAL R    CORG   MAXT   AMOUNT OF EACH TYPE OF INPUT N=5 PRODUCED
C                     MAXM
C  N5ABYM C    CORG          DOES THE AMOUNT OF N=5 PROD.VARY BY MONTH?
C  N5REQ  R    CORG   MAXI   FRACTION OF EACH TYPE OF N=5 USED / STAGE
C                     MAXT
C  QEQQ5  I    CORG          CHOOSE WHICH ALGORITHM TO COMPUTE N=5
C  QUAOFJ R    CORG   MAXJI  TO PRINT OUT AVG FEED/DAY OR INPUT=5
C                     MAXT
C  SEABYM C*1  CORG          SEA ALGAE AVAILABLE VARIES BY MONTH?
C  SEACEL R    CORG   MAXI   FRACTION OF EACH TYPE OF N=5 USED / STAGE
C              MAXM
C  ------------------------------------------------------------------
C?????????????????????????????????????????????????????????????????????


C  1. PARAMATERS
      INCLUDE (IPAR1)
C  2. ARGUMENTS
      INTEGER  RED, WRT
      LOGICAL  RERUN
C  3. COMMON BLOCKS
      INCLUDE (ICBLK1)
      INCLUDE (ICBLKO)
      INCLUDE (ICUNIT)
      INCLUDE (ICMAIN)
C  4. LOCAL VARIABLES
      CHARACTER * 1  QUEST, QUEST2
      INTEGER  UNIT, UNITN
      DIMENSION UNITN(NOOFN)

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------- ----------------------------------------------------------
C  QUEST   QUESTION TO CONTINUE WITH PROGRAM DESCRIPTION
C  QUEST2  QUESTION TO RERUN PROGRAM WITH MINOR DATA CHANGES
C  RED     ARGUE. FOR UNIT NUMBERS READ
C  RERUN   ARGUE. TO RUN AGAIN WITH A FEW CHANGES
C  WRT     ARGUE. FOR UNIT NUMBERS WRITE
C  UNIT    UNIT NO. REQUESTED FOR OUTPUT FILES
C  UNITN   UNIT NO. REQUESTED FOR SPECIFIC PROD. INPUT OUTPUT FILES
C  ------------------------------------------------------------------


      RERUN = .FALSE.

C  OPTION TO DISPLAY PROGRAM INFORMATION
      WRITE(WI,10000)  NAME
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y')  CALL INFORM

C  READ IN DATA AND CALL BODY TO COMPUTE QUANT. AND COSTS
      CALL READIN (RERUN)
100   CALL BODY

C  OPTION TO RERUN WITH A FEW CHANGES TO DATA
      WRITE(WI,30000)
      READ(RI,1000) QUEST2
C     RESPECIFY FILE READ AND WRITE UNIT NUMBERS TO INTERACTIVE
C     AND OUTPUT FILES TO NEW FILES
      IF (QUEST2 .EQ. 'Y') THEN
         WRITE(WI,49000)  DATE
         READ(RI,*) DATE
         WRITE(WI,50000)  WO
         READ(RI,*) UNIT
         WO = UNIT
         DO 200 N=1,NOOFN
            IF (OUTQN(N) .EQ. 0) THEN
               WRITE(WI,51000)  NAMEN(N), WON(N)
               READ(RI,*)  UNITN(N)
               WON(N) = UNITN(N)
            ENDIF
200      CONTINUE
         WF = WI
         RF = RI
         RERUN = .TRUE.
         CALL REREAD(WI,RI,RERUN)
         GOTO 100
      ENDIF


1000  FORMAT (A1)

10000 FORMAT (' '/////' ',15X,A,'COST SIMULATION MODEL'/' ',15X,32('-')/
     2        ///' DO YOU WANT TO SEE THE PROGRAM INFORMATION? Y OR N')

30000 FORMAT(' '//' ','DO YOU WANT TO RERUN THE PROGRAM WITH ONLY A ',
     2           'FEW CHANGES?'/' ','Y OR N'/)

49000 FORMAT(' '/' GIVE A NEW DATE OR PROGRAM IDENTIFER - UP TO 50 ',
     2       'CHARACTER IN QUOTES.'/' (PREVIOUS DATE= ',A,')')

50000 FORMAT(' '/' ','WHAT UNIT DO YOU WANT THE NEW OUTPUT FILES ',
     2       'TO GO TO? - INTEGER 10 TO 99'/' (PREVIOUS UNIT = ',I2,')')

51000 FORMAT(' ','WHAT UNIT DO YOU WANT THE ADDITIONAL INFORMATION ',
     2       'ABOUT ',A/' ','TO GO TO IF THE OPTION TO PRINT IS ',
     3       'CHOSEN'/' (PREVIOUS UNIT = ',I2,')')


      STOP 'PROGRAM COMPLETED'
      END







      SUBROUTINE INFORM
C     -----------------

C  ORGNSM:
C  -------
C  SUBROUTINE CAN CHANGE WITH ORGANISM
C  NO VARIABLES RETURNED

C  STRUCTURE: WRITES OUT INFORMATION CONCERNING PROGRAM USE

C  1. PARAMETERS
      INCLUDE (IPAR1)
C  3. COMMON BLOCKS
      INCLUDE (ICBLK1)
      INCLUDE (ICUNIT)
      INCLUDE (ICMAIN)
      INCLUDE (ICBLKO)
C  4. LOCAL VARIABLES
```

```
      INTEGER I, N, O
      CHARACTER*1  QUEST

C  NAME    DESCRIPTION OF LOCAL VARIABLES
C  ------  -------------------------------------------------------------
C  I       LOOP COUNTER FOR STAGES
C  N       LOOP COUNTER FOR INPUTS
C  O       LOOP COUNTER FOR ORGANISM
C  QUEST   TO CONTINUE
C  ---------------------------------------------------------------------


      WRITE (WI,10000) NAME, (NAMEO(O), O=1,NOOFO)
      WRITE (WI,20000) NOOFI, (NAMEI(I), I=1,NOOFI)
      WRITE(WI,25000)  MAXJI - 1
      WRITE(WI,30001)
      READ(RI,1000) QUEST

      WRITE(WI,30000) NOOFN, (NAMEN(N), NAMENU(N), N=1,NOOFN)
      WRITE(WI,31000) MAXT
      DO 32 N=1,NOOFN
         IF(OPTOK(N) .EQ. 0)  WRITE (WI,32000) NAMEN(N)
32    CONTINUE
      WRITE(WI,30001)
      READ(RI,1000) QUEST

      WRITE(WI,40000)
      DO 40 N=1,NOOFN
         DO 41 I=1,NOOFI
            IF (OPNOK(I,N) .EQ. 0)  WRITE(WI,41000) NAMEN(N), NAMEI(I)
41       CONTINUE
40    CONTINUE
      WRITE(WI,50000)
      DO 50 N=1,NOOFN
         IF (OPN$OK(N) .EQ. 0) WRITE(WI,51000) NAMEN(N)
50    CONTINUE
      WRITE(WI,60000) MAXM, NAMEM(1), NAMEM(MAXM)
      WRITE(WI,30001)
      READ(RI,1000) QUEST

      WRITE(WI,70000)
      WRITE(WI,30001)
      READ(RI,1000)  QUEST


1000  FORMAT(A1)

30001 FORMAT(' '/' CONTINUE?')

10000 FORMAT (' '//' PROGRAM IS CAPABLE OF CALCULATING COSTS FOR THE ',
     2        'FOLLOWING ',A,'TYPES:'/6(2X,A10/))

20000 FORMAT (' THE PRODUCTION PROCESS IS BROKEN INTO THE FOLLOWING ',
     2        I2,' STAGES:'/6(2X,A10/))

25000 FORMAT (' THESE STAGES CAN THEN BE BROKEN DOWN INTO SMALLER ',
     2        'UNITS OR PERIODS.'/' THE MAXIMUM NUMBER OF PERIODS ',
     3        'ALLOWED IN ALL STAGES COMBINED IS ',I2)

30000 FORMAT (' '/' THE QUANTITY OF EACH OF THE ',I2,' INPUTS USED IN ',
     2       'EVERY STAGE IS EITHER '/' ','COMPUTED OR ENTERED.  ',
     3       'THESE INPUTS ARE:'/' ',6(1X,A10,' PER ',A5/' ')/)

31000 FORMAT(' ','SOME OF THESE INPUTS MAY BE SEPARATED INTO UP TO ',
     2        I2,' TYPES'/' THE USER CAN ',
     3        'CHOOSE THE NUMBER OF TYPES OF THE FOLLOWING INPUTS:')

32000 FORMAT (2X,A10)

40000 FORMAT(' '/' THE PROGRAM IS CAPABLE OF COMPUTING THE FOLLOWING ',
     2        'INPUT QUANTITIES:')

41000 FORMAT(' ',1X,A,'IN THE ',A10,' STAGE')

50000 FORMAT(' '/' THE PROGRAM IS CAPABLE OF COMPUTING THE FOLLOWING ',
     2        'INPUT PRICES:')

51000 FORMAT(' ',1X,A10)

60000 FORMAT(' '/' NOTE: ALTHOUGH THE PROGRAM CAN COMPUTE FOR ',I2,
     2       ' MONTHS, FROM ',A9,'TO ',A10/' ','THE PRODUCTION SEASON ',
     3       'MUST RUN SEQUENTIALLY FROM THE 1ST TO THE LAST MONTH.')

70000 FORMAT(' '/' FOLLOW THESE GUIDELINES TO ENTER REQUIRED DATA:'/
     2        '   USE A DECIMAL POINT UNLESS INTEGER IS SPECIFIED.'/
     3        '   USE NO COMMAS, NO DOLLAR SIGNS.'/
     4        '   USE NO POSITIVE SIGNS;  USE NEGATIVE WHEN APPLICABLE.'/
     5        '   ALWAYS LEAVE A SPACE BETWEEN ENTRIES ON THE SAME LINE'/
     6        '   ENTER 0. IF NO VALUE EXISTS FOR THE REQUEST.'/
     7        '   USE METRIC UNITS UNLESS OTHERWISE SPECIFIED.'/
     8        '   ERRORS CAN BE CORRECTED PERIODICALLY.')


      RETURN
      END








      SUBROUTINE READIN(RERUN)
C     ------------------------

C  ORGNSM:
C  --------
C  SUBROUTINE DOES NOT VARY
C  VARIABLES DO NOT CHANGE

C  STRUCTURE:  OPTION TO INPUT BY FILE OR INTERACTIVELY
C              DISPLAY AND CHOOSE OPTIONS FOR DATA INPUT
C              READ IN DATA BY CALLING:
C                   READS
C                   READO
C                   READN, N=1,NUMBER OF INPUTS
C              2ND ENTRY POINT, REREAD, TO RERUN WITH ONLY MINOR CHANGES
C              OPTION TO CHECK, CORRECT, AND PRINT DATA

C  1. PARAMATERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS
      INTEGER  RED, RERED, WRT, REWRT, HREAD
      LOGICAL  RERUN, RUN2
C  3. COMMON BLOCKS
      INCLUDE  (ICMAIN)
      INCLUDE  (ICBLK1)
```

```
      INCLUDE  (ICBLKO)
      INCLUDE  (ICUNIT)
C  4. LOCAL VARIABLES
      INTEGER  I, INFO, K, K2, KGOTO, KLAST, N, NOOFK, OPN,
     2         R, UNIT, W, Z
      LOGICAL  CHECK, RIPROB, RUN1
      CHARACTER * 1  QFILE, QUEST, QUEST2, QUEST3
      PARAMETER  (NOOFK = 8)
      DIMENSION INFO(NOOFK)

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------  --------------------------------------------------------
C  CHECK   USED TO DETERMINE IF LOOP HAS BEEN ENTERED
C  HREAD   ARGUE: IOSTAT TO SHOW PROBLEM IN READING FILE (2-WAY COMM.)
C  I       LOOP COUNTER FOR STAGE,I
C  INFO    INFORMATION LINE NUMBER
C  K       COUNTER FOR LINE NUMBER
C  K2      COUNTER FOR SUBROUTINE LINE NUMBERS
C  KGOTO   NO. OF THE INFO LINE TO REINPUT DATA
C  KLAST   THE LAST INFO LINE,K, INPUT
C  N       LOOP COUNTER FOR INPUTS
C  NOOFK   NUMBER OF INPUT LINES
C  OPN     TO COUNT UP THE NO. OF N TO ENTER FOR EACH I
C  QFILE   WANT TO USE A DATA FILE?
C  QUEST   GENERAL QUESTION FOR USER (IS WRITTEN OVER)
C  QUEST2  OPTION TO STOP, REDO, OR CONTINUE WITH DATA ENTRY
C  QUEST3  OPTION TO SEE OPTION INFORMATION
C  R       UNIT NUMBER FOR READ
C  RED     ARGUE:  UNIT NUMBERS FOR READ
C  RERED   ARGU:  UNIT NO. FOR 2ND ENTRY POINT - ALWAYS INTERACTIVE
C  RERUN   ARGU:  TO ALLOW INFO PRINTING IF NOT RUN1, FROM 2ND ENTRY PNT
C  REWRT   ARGUE:  UNIT NO FOR 2ND ENTRY POINT - ALWAYS INTERACTIVE
C  RIPROB  TO DETERMINE IF FILE PROBLEM HAS BEEN ENCOUNTERED
C  RUN1    USED TO PRINT INFO THE FIRST RUN THROUGH
C  RUN2    ARGUE: IF FIRST RUN THROUGH DATA INPUT ALREADY DONE
C  UNIT    UNIT NO FOR OUTPUT FILE  TO GO TO
C  W       UNIT NUMBER FOR WRITE
C  WRT     ARGUE:  UNIT NUMBERS FOR WRITE
C  Z       2ND COUNTER FOR LINE NUMBERS
C  -------------------------------------------------------------------


C  SET ALL LINES TO 0 IN ORDER TO READ IN THE FIRST TIME
      DO 100 K=1,NOOFK
         INFO(K) = 0
100   CONTINUE
      RUN1 = .TRUE.
      RUN2 = .FALSE.

C  INITIALIZE FILE VARIABLES
      RIPROB = .FALSE.
      HREAD  = 0

C  OPTION TO TERMINATE
      WRITE(WI,20000)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y')  STOP 'COMPLETE SPREAD SHEET'

C  READ IN PROGRAM DATA/IDENTIFIER AND OUTPUT FILE UNIT NUMBER
      WRITE(WI,25000)
      READ(RI,*) DATE
      WRITE(WI,26000)
      READ(RI,*) UNIT
      WO = UNIT


C  OPTION TO INPUT BY SEQUENTIAL FILE
      WRITE(WI,30000)
      READ(RI,1000) QFILE
      IF (QFILE .EQ. 'Y') THEN
         WRITE(WI,31000)
         READ(RI,*) NAMEF1
         WRITE(WI,32000)
         W = WF
         R = RF
         WRT = WF
         RED = RF

      ELSE
         W = WI
         R = RI
         WRT = WI
         RED = RI
         NAMEF1 = 'NOFILE'
      ENDIF

C  OPTION INFORMATION IDENTIFIER
      WRITE(WI,39000)


C  LIST OPTIONS AVAILABLE TO READ DATA IF OPTION TO DISPLAY CHOSEN
250   WRITE(W,40000)
      WRITE(W,40010)
      READ(R,1000) QUEST3
      IF (QUEST3 .EQ. 'Y') THEN
         IF (OPQOK(1) .EQ. 0) THEN
            WRITE(W,40100)
            WRITE(W,40150)
         ENDIF
         IF (OPQOK(2) .EQ. 0) THEN
            WRITE(W,40200)
            WRITE(W,40150)
         ENDIF
         IF (OPQOK(3) .EQ. 0)  WRITE(W,40300)
         WRITE(W,40500)
         READ(R,1000,ERR=600,IOSTAT=HREAD) QUEST

C        LIST INPUTS CALCULATABLE IN EACH STAGE
         WRITE(W,41000)
         DO 300 N=1,NOOFN
            WRITE(W,41100) N, NAMEN(N)
            CHECK = .TRUE.
            DO 310 I=1,NOOFI
               IF (OPNOK(I,N) .EQ. 0) THEN
                  WRITE(W,41200) NAMEI(I)
                  CHECK = .FALSE.
               ENDIF
310         CONTINUE
            IF (CHECK)  WRITE(W,41300)
300      CONTINUE
         WRITE(W,40500)
         READ(R,1000,ERR=600,IOSTAT=HREAD) QUEST
      ENDIF


C  USER CHOOSES AN OPTION
      WRITE(W,42000)
      READ(R,*,ERR=600,IOSTAT=HREAD) OPTION
```

```
C  OPTION 1 REQUIRES DEFINITION OF THE FIRST STAGE OF INTEREST
      IF (OPTION .EQ. 1) THEN
320      WRITE(W,43000)
         READ(R,*,ERR=600,IOSTAT=HREAD) BEGINI
         IF (BEGINI .LE. 1  .OR.  BEGINI .GT. NOOFI) THEN
            WRITE(W,43100)
            GOTO 320
         ENDIF
      ELSE
         BEGINI = 1
      ENDIF

C  USER CHOOSES WHICH INPUTS TO READ IN AS QUANTITY ONLY, INiTIALIZE TO
C  1 ONLY THOSE THAT ARE COMPUTABLE ARE OFFERED
      IF (OPTION .NE. 3) THEN
         WRITE(W,44000)
         DO 350 N=1,NOOFN
            WRITE(W,41100) N, NAMEN(N)
            CHECK = .TRUE.
            DO 360 I=1,NOOFI
               IF (I .LT. BEGINI) THEN
                  OPNNEW(I,N) = '-'
               ELSEIF (OPNOK(I,N) .EQ. 0  .AND.  I .GE. BEGINI) THEN
                  WRITE(W,41200) NAMEI(I)
                  READ(R,1000,ERR=600,IOSTAT=HREAD) OPNNEW(I,N)
                  CHECK = .FALSE.
               ELSEIF (OPNOK(I,N) .NE. 0  .AND.  I .GE. BEGINI) THEN
                  OPNNEW(I,N) = 'E'
               ENDIF
360         CONTINUE
            IF (CHECK) WRITE(W,41300)
350      CONTINUE
      ENDIF

C  IF NO STAGES TO BE COMPUTED FOR AN INPUT,
C  THEN ANYNOK(N) = TOTAL STAGES TO COMPUTE
      DO 370 N=1,NOOFN
         ANYNOK(N) = 0
         DO 380 I=BEGINI,NOOFI
            IF (OPNNEW(I,N) .EQ. 'C') THEN
               OPN = 0
            ELSEIF (OPNNEW(I,N) .EQ. 'E') THEN
               OPN = 1
            ENDIF
            ANYNOK(N) = ANYNOK(N) + OPN
380      CONTINUE
370   CONTINUE
      TOTALI = NOOFI - BEGINI + 1


C  OPTION TO CHECK AND REINPUT OPTION INFORMATION
      WRITE (WI,45000)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y')  CALL ROUT(WI)
      IF (NAMEF1 .EQ. 'NOFILE') THEN
         WRITE(WI,45100)
         READ(RI,1000) QUEST
         IF (QUEST .EQ. 'Y')  GOTO 250
      ENDIF

C  OPTION TO PRINT OPTION INFORMATION
      WRITE(WI,45300)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y')  CALL ROUT(WO)

C  OFFER OPTION TO TERMINATE
      WRITE(WI,45200)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y')  STOP 'PROGRAM STOPPED IN OPTION SECTION'

C  DISPLAY NECESSARY SUBROUTINES AND THEIR NUMBERS
400   IF (RUN1)  THEN
         WRITE(W,50000)
         WRITE(W,50010)
         K2 = 1
         WRITE(W,50020) K2
         K2 = K2+1
         WRITE(W,50030) K2, NAME
         DO 410 N=1,NOOFN
            K2 = K2+1
            IF(ANYNOK(N) .LT. TOTALI)  WRITE(W,50040) K2, NAMEN(N)
410      CONTINUE
      ENDIF
      HREAD = 0
      RIPROB = .FALSE.


C  READ IN DATA NECESSARY TO RUN GENERAL PROGRAM
C  CONTROL TRANSFERED HERE IF INPUT CORRECTIONS ARE TO BE MADE
420   CONTINUE
      K = 1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(WI,51000) K
         CALL READS(WRT,RED,HREAD,RUN2)
         IF (HREAD .NE. 0)  GOTO 550
      ENDIF
      K = K + 1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(WI,51100) K, NAME
         CALL READO(WRT,RED,HREAD,RUN2)
         IF (HREAD .NE. 0) GOTO 550
      ENDIF

C  READ IN DATA PERTAINING TO QUANTITY OF INPUT, N, ONLY IF
C  SOME STAGE IS TO BE COMPUTED FOR THAT INPUT
      DO 500 N=1,NOOFN
         K = K+1
         IF (INFO(K) .EQ. 0  .AND.  ANYNOK(N) .LT. TOTALI) THEN
            WRITE(WI,53000) K, NAMEN(N)
            IF (N .EQ. 1)  CALL READN1(WRT,RED,HREAD,RUN2)
            IF (N .EQ. 2)  CALL READN2(WRT,RED,HREAD,RUN2)
            IF (N .EQ. 3)  CALL READN3(WRT,RED,HREAD,RUN2)
            IF (N .EQ. 4)  CALL READN4(WRT,RED,HREAD,RUN2)
            IF (N .EQ. 5)  CALL READN5(WRT,RED,HREAD,RUN2)
            IF (N .EQ. 6)  CALL READN6(WRT,RED,HREAD,RUN2)
            IF (HREAD .NE. 0)  GOTO 550
         ENDIF
500   CONTINUE

      WRITE(WI,59000)
550   KLAST = K


C  CONTROL TRANSFERED HERE IF ERROR FOUND IN READING FILE
600   IF (QFILE .EQ. 'Y') THEN
         KLAST = K
C        MESSAGE IF ERROR DETECED IN READING FILE
```

```
      IF (HREAD .LT. 0)  WRITE(WI,60000)
      IF (HREAD .GT. 0)  WRITE(WI,61000)
      IF (HREAD .NE. 0)  RIPROB = .TRUE.
      QFILE = 'N'
      HREAD = 0
      W = WI
      R = RI
      WRT = WI
      RED = RI
   ENDIF


C  RETURN POINT FROM MAIN PROGRAM TO RERUN THE PROGRAM WITH FEW CHANGES
   ENTRY REREAD(REWRT,RERED,RERUN)
   WRT = WI
   RED = RI
   W   = WI
   R   = RI


C  OPTION TO CHECK DATA THAT HAS BEEN INPUT
   WRITE(WI,62000)
   READ(RI,1000) QUEST
   IF (QUEST .EQ. 'Y') THEN
      WRITE(WI,62100)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y')  CALL ROUT(WI)
      Z = 1
      WRITE(WI,62200)  Z
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y')  CALL ROUTS(WI)
      Z = Z + 1
      WRITE(WI,62210)  NAME, Z
      READ(RI,1000)  QUEST
      IF (QUEST .EQ. 'Y')  CALL ROUTO(WI)
      DO 605 N=1,NOOFN
         Z = Z + 1
         IF (ANYNOK(N) .LT. TOTALI)   THEN
            WRITE(WI,62300)  Z, NAMEN(N)
            READ(RI,1000) QUEST
            IF (QUEST .EQ. 'Y') THEN
               IF (N .EQ. 1)  CALL ROUTN1(WI)
               IF (N .EQ. 2)  CALL ROUTN2(WI)
               IF (N .EQ. 3)  CALL ROUTN3(WI)
               IF (N .EQ. 4)  CALL ROUTN4(WI)
               IF (N .EQ. 5)  CALL ROUTN5(WI)
               IF (N .EQ. 6)  CALL ROUTN6(WI)
            ENDIF
         ENDIF
605   CONTINUE
   ENDIF

C  OPTION TO TERMINATE, REDO INTERACTIVELY, OR CONTINUE
C     PROBLEM WITH FILE, OFFER A DIFFERENT SET OF CHOICES
610   IF (RIPROB) THEN
      WRITE(WI,63000)
      WRITE(WI,63100)
   ELSEIF (.NOT. RIPROB) THEN
      WRITE(WI,63000)
      WRITE(WI,63200)
   ENDIF

C     CONTROL TRANSFERED ACCORDING TO REQUEST
C     AND INPUT LINES ARE SET TO 0 OR 1 ACCORDINGLY
      READ(RI,1000) QUEST2
      IF ((QUEST2.NE.'S') .AND. (QUEST2.NE.'R') .AND. (RIPROB)) THEN
         GOTO 610
      ELSEIF (QUEST2 .EQ. 'S' )  THEN
         GOTO 800
      ELSEIF (QUEST2 .EQ. 'R' )  THEN
         RIPROB = .FALSE.
         NAMEF1 = 'NOFILE'
         RUN1 = .TRUE.
         RUN2 = .FALSE.
         W = WI
         R = RI
         DO 620 K=1,NOOFK
            INFO(K) = 0
620      CONTINUE
         WRT = WI
         RED = RI
         GOTO 250
      ELSEIF (QUEST2 .EQ. 'C') THEN
         DO 700 K=1,NOOFK
            INFO(K) = 1
700      CONTINUE
      ENDIF

      RUN1 = .FALSE.
      RUN2 = .TRUE.
      RERUN = .FALSE.


C  OPTION TO CORRECT LINE BY CHOOSING A LINE TO CORRECT, EXCEPT OPTIONS
710   WRITE(WI,70000) NOOFK
      READ(RI,*) KGOTO
      IF (KGOTO .LT. 0  .OR.  KGOTO .GT. NOOFK) THEN
         GOTO 710
      ELSEIF (KGOTO .GT. 0) THEN
         INFO(KGOTO) = 0
         GOTO 400
      ENDIF

C  OPTION TO PRINT DATA ENTERED
800   WRITE(WI,80000)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y') THEN
         CALL ROUT(WO)
         CALL ROUTS(WO)
         CALL ROUTO(WO)
         IF (ANYNOK(1) .LT. TOTALI)  CALL ROUTN1(WO)
         IF (ANYNOK(2) .LT. TOTALI)  CALL ROUTN2(WO)
         IF (ANYNOK(3) .LT. TOTALI)  CALL ROUTN3(WO)
         IF (ANYNOK(4) .LT. TOTALI)  CALL ROUTN4(WO)
         IF (ANYNOK(5) .LT. TOTALI)  CALL ROUTN5(WO)
         IF (ANYNOK(6) .LT. TOTALI)  CALL ROUTN6(WO)
      ENDIF

      IF(QUEST2.EQ.'S') STOP 'DATA ENTRY HALTED '


1000  FORMAT (A1)

20000 FORMAT (' '//' ','COMPLETE THE PROGRAM DOCUMENTATION SPREAD',
     2             ' SHEET BEFORE CONTINUING.'/
     3         ' ','DO YOU WANT TO TERMINATE TO COMPLETE THIS FORM?  ',
     4             'Y OR N')
```

```
25000 FORMAT(' '//' ENTER THE DATE OR A PROGRAM IDENTIFIER - ',
     2        'UP TO 50 CHARACTERS IN SINGLE QUOTES')

26000 FORMAT(' '/' TO WHAT UNIT SHOULD THE OUTPUT FILES BE SENT?',
     2        ' - INTEGER 10 TO 99')

30000 FORMAT (' '/' ','DATA CAN BE ENTERED INTERACTIVELY OR BY FILE.'/
     2        ' ','DO YOU WANT TO USE AN EXISTING DATA FILE?  ',
     3           'Y OR N')

31000 FORMAT (' '/' ENTER THE NAME OF THE DATA FILE - ',
     2        'UP TO 10 CHARACTERS IN SINGLE QUOTES')

32000 FORMAT(' '/' ','THE NAME OF EACH SECTION CURRENTLY BEING READ ',
     2       'FROM THE FILE WILL APPEAR. '/' YOU WILL ',
     3       'THEN HAVE THE OPPORTUNITY TO LOOK AT AND TO CHANGE ',
     4       'THE DATA.')

39000 FORMAT(' '////' OPTION INFORMATION (MAY NOT BE CHANGED IF FILE ',
     2       'USED)'/' ',51('-'))

40000 FORMAT (' '//' YOU HAVE SEVERAL OPTIONS FOR ENTERING DATA:'/
     2        ' ','CHOOSE AN OPTION THAT IS COMPATABLE WITH THE PRO',
     3           'GRAMS CAPABILITIES AND'/' ','WITH THE DATA THAT IS ',
     4           'AVAILABLE TO YOU, THE USER.'/
     5        ' ','REFER TO THE PROGRAM DOCUMENTATION FOR THE ',
     6           'DATA THAT WILL BE REQUIRED.')

40010 FORMAT (' '/' ','DO YOU WANT TO SEE THE INFORMATION CONCERNING ',
     2        'THE OPTION INFORMATION?  Y OR N')

40100 FORMAT(' '/' ','OPTION 1:'/' ',9('-')/' ','YOU MUST CHOOSE AT WH',
     2           'ICH STAGE TO BEGIN COMPUTATIONS.'/' ','FOR THOSE ',
     3           'STAGES BEFORE THE FIRST STAGE OF INTEREST, YOU ',
     4           'MUST ENTER'/' ','THE TOTAL COST AND ',
     5           'ELAPSED TIME FOR THOSE STAGES NOT COMPUTED INDIVID',
     6           'UALLY.'/' ','FOR THOSE STAGES AFTER THE FIRST STA',
     7           'GE OF INTEREST, YOU HAVE A CHOICE;')

40150 FORMAT(' ','FOR EACH INPUT IN EACH STAGE, YOU MAY CHOOSE TO ENT',
     2           'ER THE QUANTITY'/' ','OF THAT INPUT USED IN A STAGE ',
     3           'OR YOU MAY CHOOSE TO ALLOW'/' ','THE PROGRAM TO COM',
     4           'PUTE THIS QUANTITY')

40200 FORMAT(' '/' ','OPTION 2:'/' ',9('-'))

40300 FORMAT(' '/' ','OPTION 3:'/' ',9('-')/' ',3X,'PROGRAM COMPUTES',
     2           ' THE QUANTITY OF ALL INPUTS USED IN EVERY STAGE.')

40500 FORMAT(' '/' ','CONTINUE?')

41000 FORMAT(' ',//' ','THE QUANTITY OF INPUT USED CAN BE COMPUTED BY ',
     2           'THE PROGRAM'/' FOR THE FOLLOWING STAGES:')

41100 FORMAT(' '/' ','INPUT ',I2,', ',1X,A10,':')

41200 FORMAT(' ',2X,A10)

41300 FORMAT(' ',2X,'NO STAGES ARE COMPUTABLE FOR THIS INPUT')

42000 FORMAT(' '/' ','NOW, CHOOSE ONE OF THE ABOVE OPTIONS BY ',
     2                'NUMBER.')

43000 FORMAT(' '/' ','YOU HAVE CHOSEN OPTION 1. AT WHICH STAGE DO YOU ',
     2                'WANT TO BEGIN COMPUTATION?')

43100 FORMAT(' '/' ','FOR OPTION 1, FIRST STAGE MUST BE GREATER THAN 1',
     2        ' ','AND NO GREATER THAN THE NUMBER OF STAGES. PLEASE ',
     3            'REINPUT')

44000 FORMAT(' '/' ','FOR EACH INPUT, YOU CAN CHOOSE TO ENTER THE ',
     2            'QUANTITY OF AN INPUT USED, '/' ','OR YOU CAN CHOOSE ',
     3           'TO LET THE PROGRAM COMPUTE THE QUANTITIES.'/
     4        ' ','(STAGES NOT LISTED MUST BE ENTERED AS QUANTITIES.)'//
     5        ' ',2X,'ENTER  C  TO LET THE PROGRAM COMPUTE THE QUANTITY',
     6            ' IN A STAGE OR'/' ',2X,'ENTER  E  TO ENTER YOUR OWN ',
     7            'QUANTITY.')

45000 FORMAT(' '/' ','DO YOU WANT TO SEE THE OPTION INFORMATION? ',
     2        ' Y OR N')

45100 FORMAT(' '//' ','DO YOU WANT TO CHANGE ANY OF THIS INFORMATION? '/
     2        ' ','IF YES, THEN ALL INFORMATION MUST BE CHANGED')

45200 FORMAT(' '/' DO YOU WANT TO HALT THE PROGRAM AT THIS POINT?')

45300 FORMAT(' '/' DO YOU WANT TO PRINT THE OPTION INFORMATION?')

50000 FORMAT(' '//' ','YOU WILL NOW BE ASKED TO ENTER ALL DATA ',
     2            'NECESSARY TO RUN THIS PROGRAM'/
     3        ' ','EACH SECTION BEGINS A NEW SET OF INPUT LINE NUMBERS.'/
     4        ' ','FOLLOW THE PD. SPREAD SHEET FOR THE LINE NUMBERS ',
     5        ' OF EACH SECTION')

50010 FORMAT (' '/' DATA MUST BE ENTERED FOR THE FOLLOWING SECTIONS:')

51000 FORMAT(' '///' ','SECTION ',I1,':',2X,'GENERAL DATA'/' ',25('-'))

50020 FORMAT(' SECTION ',I1,':',2X,'GENERAL    DATA')

50030 FORMAT(' SECTION ',I1,':',2X,A10,' DATA')

50040 FORMAT(' SECTION ',I1,':',2X,A10,' DATA')

51100 FORMAT(' '///' ','SECTION ',I1,':',2X,A,' DATA'/' ',27('-'))

53000 FORMAT(' '///' ','SECTION ',I2,':',2X,A10,1X,'DATA'/' ',27('-'))

59000 FORMAT(' '//' ******* DATA ENTRY COMPLETED *******')

60000 FORMAT(' ','END OF FILE HAS BEEN REACHED BEFORE DATA ENTRY ',
     2            'COMPLETED.'/
     3        ' ','PROGRAM CAN NOT CONTINUE WITHOUT CORRECTIONS'////)

61000 FORMAT(' ','AN ERROR WAS DETECTED IN THE DATA FILE.'/
     2        ' ','PROGRAM CAN NOT CONTINUE WITHOUT CORRECTIONS')

62000 FORMAT(' '//' DATA ENTERED IN ANY SECTION CAN NOW BE CHECKED',
     2            ' FOR ERRORS, ONE AT A TIME.'/
     3        ' ','DO YOU WANT TO SEE ANY OF THE ENTERED DATA?')

62100 FORMAT(' '/' ','DATA CONCERNING OPTIONS?  (CAN NOT BE CHANGED)')

62200 FORMAT(' '/' ','DATA FOR GENERAL SECTION ',I2,'?')

62210 FORMAT(' '/' ','DATA FOR ',A,' SECTION ',I2,'?')

62300 FORMAT(' '/' ','DATA FOR SECTION ',I2,' CONCERNING ',A,'?')
```

```
63000 FORMAT(' ',/' ','YOU NOW HAVE SEVERAL OPTIONS CONCERNING ALL ',
     2              'THE DATA ENTERED.  YOU CAN:'/
     3      ' ',2X,'ENTER S TO STOP AFTER AN OPTION TO PRINT THE DATA'/
     3      ' ',2X,'ENTER R TO REDO DATA ENTIRELY INTERACTIVELY')

63100 FORMAT(' ','ENTER S OR R')

63200 FORMAT(' ',2X,'ENTER C TO CONTINUE EXECUTION WITH CORRECTIONS'/
     2       ' ','ENTER S, R, OR C')

70000 FORMAT (' '/' YOU MAY NOW CORRECT ENTERED DATA ONE SECTION ',
     2          'AT A TIME'/
     3         ' ',2X,'ENTER 0 TO MAKE NO CORRECTIONS, OR'/
     4         ' ',2X,'ENTER THE NUMBER OF THE SECTION IN WHICH YOU ',
     5              'WISH TO MAKE CORRECTIONS'/
     6         ' ','ENTER AN INTEGER 0 TO',1X,I2)

80000 FORMAT (' '/' ','DO YOU WANT A PRINTED COPY OF ALL OF THE DATA ',
     2         ' ',' ENTERED?  Y OR N.')


      RETURN
      END





      SUBROUTINE ROUT(WRT)
C     --------------------

C  ORGNSM:
C  ------
C  SUBROUTINE DOES NOT VARY
C  VARIABLES DO NOT CHANGE

C  STRUCTURE :  DISPLAYS OR PRINTS OUT DATA FROM SUBROUTINE READIN

C  1. PARAMATERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS
      INTEGER WRT
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICBLK1)
      INCLUDE (ICBLKO)
      INCLUDE (ICUNIT)
C  1. LOCAL VARIABLES AND ARGUMENTS
      INTEGER I, K, N, W

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------  -------------------------------------------------------------
C  I       LOOP COUNTER FOR STAGES
C  K       INPUT LINE NUMBER
C  N       LOOP COUNTER FOR INPUTS
C  W       UNIT NUMBERS FOR WRITE
C  WRT     ARGUE:  UNIT NUMBER FOR WRITE
C  ------------------------------------------------------------------


C  TRANSLATE ARGUMENTS TO LOCAL VARIABLES
      W = WRT

C  DISPLAY OR PRINT ALL DATA ENTERED IN SUBROUTINE READIN
C  HEADING FOR OUTPUT
      WRITE(W,10000) NAME, DATE
      IF (NAMEF1 .NE. 'NOFILE')  WRITE(W,20000) NAMEF1


C  OPTION INFORMATION
      WRITE(W,30000) OPTION
         IF (OPTION .EQ. 1)  WRITE(W,31000)
         IF (OPTION .EQ. 2)  WRITE(W,32000)
         IF (OPTION .EQ. 3)  WRITE(W,33000)

      IF (OPTION .EQ. 1)  WRITE(W,34000) NAMEI(BEGINI)

      IF (OPTION .EQ. 1  .OR.  OPTION .EQ. 2) THEN
         WRITE(W,40000) (NAMEI(I), I=1,NOOFI)
         DO 100 N=1,NOOFN
            WRITE(W,41000) NAMEN(N), (OPNNEW(I,N), I=1,NOOFI)
100      CONTINUE
      ENDIF


C  THE NUMBER AND NAME OF EACH INPUT SUBROUTINE
      K = 1
      WRITE(W,45000)
      WRITE(W,50000) K
      K=K+1
      WRITE(W,51000) K, NAME
      DO 200 N=1,NOOFN
         K = K+1
         IF(ANYNOK(N) .LT. TOTALI)  WRITE(W,52000) K, NAMEN(N)
200   CONTINUE


10000 FORMAT('1',A10,' COST SIMULATION MODEL:  INPUTS ENTERED'/
     2       ' ',49('-')/
     3       ' ','DATE:  ',A50/' ','OPTION DATA ENTERED'/)

20000 FORMAT(' ','NAME OF FILE USED:',2X,A10)

30000 FORMAT(' ','OPTION CHOSEN:',1X,I1)

31000 FORMAT(' ',15X,'COST UP TO FIRST STAGE OF INTEREST IS ENTERED'/
     2     ' ',15X,'USER CHOOSES INPUTS TO COMPUTE IN REMAINING STAGES')

32000 FORMAT(' ',15X,'USER CHOOSES WHICH INPUTS TO COMPUTE IN EACH ',
     2                'STAGE')

33000 FORMAT(' ',15X,'ALL INPUTS IN ALL STAGES ARE COMPUTED ',
     2                'BY THE PROGRAM')

34000 FORMAT(' ',15X,'THE FIRST STAGE OF INTEREST IS THE',1X,A10,
     2                'STAGE')

40000 FORMAT(' '/' ','IF C, THE PROGRAM COMPUTES THE QUANTITY'/
     2            ' ','IF E, THE USERS ENTERS THE QUANTITY'/
     3            ' ','IF -, THE QUANTITY IS NEITHER ENTERED ',
     4                'NOR COMPUTED'//
     5       ' ','INPUT',T15,7(A10,1X))

41000 FORMAT(' ',A10,T15,7(A1,10X))

45000 FORMAT(' '/' ','THE NUMBER OF EACH SECTION FOR WHICH DATA ',
```

```
     2              'MUST BE ENTERED.')

50000 FORMAT(' ',2X,'SECTION',1X,I2,': GENERAL    DATA')

51000 FORMAT(' ',2X,'SECTION',1X,I2,': ',A,' DATA')

52000 FORMAT(' ',2X,'SECTION',1X,I2,':',1X,A10,1X,'DATA')


      RETURN
      END








      SUBROUTINE READS(WRT,RED,HREAD,RUN2)
C     -----------------------------------

C  ORGNSM:
C  ------
C  SUBROUTINE DOES NOT VARY
C  VARIABLES DO NOT CHANGE
C  **NOTE:  SOME INFO(K) MAY HAVE TO BE FORCED TO 0 DEPENDING ON HOW
C           ORGANISM DEPENDENT SUBROUTINES ARE MANAGED

C  STRUCTURE: READS IN GENERAL DATA TO RUN PROGRAM
C             READS IN PRICE OF EACH INPUT OR ELSE CALLS PRICE, N, TO
C                COMPUTE PRICE
C             OPTION TO CHECK, CORRECT, OR PRINT DATA

C  1. PARAMETERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS
      INTEGER  HREAD, RED, RED2, WRT, WRT2, WW
      LOGICAL  RUN2, RUN3
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICBLK1)
      INCLUDE  (ICBLKO)
      INCLUDE (ICUNIT)
C  4. LOCAL VARIABLES
      INTEGER  I, INFO, K, K2, KGOTO, KLAST, M, N, NOOFK, R, T, W
      LOGICAL  CHECK, RUN1
      CHARACTER * 1  QUEST, QUEST2
      PARAMETER  (NOOFK = 54)
      DIMENSION  INFO(NOOFK)

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------  ---------------------------------------------------------
C  CHECK   TO DETERMINE IF A LOOP HAS BEEN ENTERED
C  HREAD   ARGUE:  ISOSTAT TO INDICATE PROBLEMS IN READING FILE
C  I       LOOP COUNTER FOR STAGES
C  INFO    INFORMATION LINE NUMBER
C  K       NUMBER OF THE INFO LINE
C  K2      LOOP COUNTER TO FORCE REINPUT
C  KGOTO   LINE NUMBER TO CORRECT
C  KLAST   LAST INFO LINE NUMBER TO BE INPUT
C  M       LOOP COUNTER FOR MONTH
C  N       LOOP COUNTER FOR INPUT
C  NOOFK   NUMBER OF INPUT LINES
C  QUEST   GENERAL VARIABLE FOR USER QUESTION (IS WRITTEN OVER)
C  QUEST2  OPTION TO STOP, REDO, OR CONTINUE DATA ENTRY
C  R       UNIT NUMBER FOR READ
C  RED     ARGUE:  FOR UNIT NUMBERS TO READ
C  RED2    ARGUE:  UNIT NO. TO READ FOR NEXT LEVEL SUBR.
C  RUN1    FIRST RUN THROUGH
C  RUN2    ARGUE:  INDICATES IF ISNT FIRST RUN THROUGH
C  RUN3    ARGUE:  INDICATES TO NEXT LEVEL SUBROUTINE NOT 1ST RUN THRU
C  T       LOOP COUNTER FOR TYPE OF INPUT,I
C  W       UNIT NUMBER FOR WRITE
C  WRT     ARGUE:  FOR UNIT NUMBERS TO WRITE
C  WRT2    ARGUE:  UNIT NO. TO WRITE FOR NEXT LEVEL SUBR.
C  WW      ARGUE:  UNIT NO. TO WRITE AS OUTPUT FILE
C  -----------------------------------------------------------------


C  NOT FIRST RUN, IMMEDIATELY OFFER OPTION TO CHECK AND CHANGE DATA
      IF (RUN2) THEN
         DO 120  K=1,NOOFK
            INFO(K) = 1
120      CONTINUE

         WRT2 = WI
         RED2 = RI
         W    = WI
         R    = RI
         RUN1 = .FALSE.
         RUN3 = .TRUE.

         WRITE(WI,60000)
         READ(RI,1000)  QUEST
         IF (QUEST .EQ. 'Y')  CALL ROUTS(WI)

100      WRITE(WI,70000) NOOFK
         READ(RI,*) KGOTO
         IF (KGOTO .LT. 0  .OR.  KGOTO .GT. NOOFK) THEN
            GOTO 100
         ELSEIF (KGOTO .GT. 0) THEN
            INFO (KGOTO) = 0
         ELSEIF (KGOTO .EQ. 0)  THEN
            GOTO 850
         ENDIF

      ELSE
C     INITIALIZE INPUT LINES TO READ IN ALL DATA THE FIRST RUN
C     ASSIGN ARGUE. TO LOCAL UNIT NUMBERS TO BE CONSISTENT W/ OPTION
         W = WRT
         R = RED
         WRT2 = WRT
         RED2 = RED
         DO 110 K=1,NOOFK
            INFO(K) = 0
110      CONTINUE
         RUN1 = .TRUE.
         RUN3 = .FALSE.
      ENDIF


C  READ IN DATA NECESSARY TO RUN THE GENERAL PROGRAM
400   HREAD = 0

C  BEGINNING AND ENDING MONTHS OF PRODUCTION
      K = 1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,21000) K
```

```
         READ(R,*,ERR=900,IOSTAT=HREAD) BEGINM
C        FORCE ALL INFO(K) TO 0 IN CASE OF REINPUT
         DO 405 K2 = 2,NOOFK
            INFO(K2) = 0
405      CONTINUE
         RUN2 = .FALSE.
      ENDIF
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,22000) K
         READ(R,*,ERR=900,IOSTAT=HREAD) ENDM
C        FORCE ALL REMAINING INFO(K) TO 0 IN CASE OF REINPUT
         DO 406 K2 = 3,NOOFK
            INFO(K2) = 0
406      CONTINUE
         RUN2 = .FALSE.
      ENDIF

C  DETERMINE THE LENGTH OF STAGES,I, AND THE NO. OF PERIODS,J, IN EACH I
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,23000) K
         DO 410 I=BEGINI,NOOFI
            WRITE(W,23100) NAMEI(I)
            READ(R,*,ERR=900,IOSTAT=HREAD) DAYSI(I)
410      CONTINUE
      ENDIF
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,24000) K
         DO 420 I=BEGINI,NOOFI
            WRITE(W,23100) NAMEI(I)
            READ(R,*,ERR=900,IOSTAT=HREAD) JPERI(I)
420      CONTINUE
      ENDIF

C  DAYS AND COST UP TO THE END OF OPTION 1
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         IF (OPTION .EQ. 1) THEN
            WRITE(W,25000) K, NAMEI(BEGINI)
            READ(R,*,ERR=900,IOSTAT=HREAD) DAYOP1
         ENDIF
      ENDIF
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         IF (OPTION .EQ. 1) THEN
            WRITE(W,26000) K, NAMEI(BEGINI)
            READ(R,*,ERR=900,IOSTAT=HREAD) CSOP1
         ENDIF
      ENDIF

C  LOOP THROUGH EACH INPUT,N, TO OFFER PRICE OPTION
      IF (RUN1)  WRITE(W,50000)
      DO 500 N=1,NOOFN
         IF(RUN1)  WRITE(W,50100) NAMEN(N)
C        IF THE OPTION EXISTS TO COMPUTE INPUT PRICE, OFFER IT
         K = K+1
         IF (INFO(K) .EQ. 0) THEN
            IF (OPN$OK(N) .EQ. 0) THEN
               WRITE(W,51000) K, NAMEN(N)
               READ(R,1000,ERR=900,IOSTAT=HREAD) QOPN$(N)
            ELSE
               QOPN$(N) = 'N'
               WRITE(W,51100)  K, NAMEN(N)
            ENDIF
C           SET RELEVANT INFO LINES TO 0 TO FORCE REINPUTTING
            IF (QOPN$(N) .EQ. 'Y') THEN
               INFO(K+5*NOOFN) = 0
            ELSEIF (QOPN$(N) .EQ. 'N') THEN
               CONTINUE
            ENDIF
         ENDIF
500   CONTINUE

C  LOOP THRU EACH INPUT,N,  TO ENTER INFO IF OPTION TO COMPUTE $ = NO
      IF (RUN1)  WRITE(W,30000) NOOFN
      DO 600 N=1,NOOFN

C        NUMBER OF TYPES,T, OF EACH INPUT,N
         K = K+1
         IF (INFO(K).EQ.0 .AND. OPTOK(N).EQ.0 .AND. QOPN$(N).EQ.'N')THEN
            WRITE(W,31000) K, NAMEN(N)
            READ(R,*,ERR=900,IOSTAT=HREAD) NOOFT(N)
C           SET RELEVANT INFO LINES TO 0 TO FORCE REINPUT
            INFO(K+1) = 0
            INFO(K+3) = 0
C           NEXT FORCING DOESNT WORK. MAKE COMMENT FOR NOW
C           INFO(K+5*NOOFN) = 0
         ENDIF

C        NAME OF EACH TYPE,T
         K = K+1
         IF(INFO(K).EQ.0 .AND. OPTOK(N).EQ.0 .AND. QOPN$(N).EQ.'N') THEN
            IF (NOOFT(N) .GT. 1) THEN
               DO 610 T=1,NOOFT(N)
                  WRITE(W,32000) K, T, NAMEN(N)
                  READ(R,*,ERR=900,IOSTAT=HREAD) NAMET(N,T)
610            CONTINUE
            ELSE
               NAMET(N,1) = 'ONE'
            ENDIF
         ENDIF

C        DOES THE PRICE VARY BY MONTH?
         K = K+1
         IF (INFO(K) .EQ. 0  .AND.  QOPN$(N) .EQ. 'N') THEN
            IF ((ENDM-BEGINM) .GT. 0) THEN
               WRITE(W,53000) K, NAMEN(N)
               READ(R,1000,ERR=900,IOSTAT=HREAD) N$BYM(N)
C              SET RELEVANT INFO LINES TO 0 TO FORCE REINPUT
               INFO(K+1) = 0
            ELSE
               N$BYM(N) = '-'
            ENDIF
         ENDIF

C        PRICE OF EACH TYPE,T, IN ALL PRODUCTION MONTHS
         K = K+1
         IF (INFO(K) .EQ. 0  .AND.  QOPN$(N) .EQ. 'N') THEN
            WRITE(W,54000) K, NAMEN(N), NAMENU(N)
            DO 620 T=1,NOOFT(N)
               IF (NOOFT(N) .GT. 1)  WRITE(W,55000) NAMET(N,T)
               IF (N$BYM(N) .EQ. 'Y') THEN
                  DO 630 M=BEGINM,ENDM
                     WRITE(W,56000) NAMEM(M)
                     READ(R,*,ERR=900,IOSTAT=HREAD) PRICET(N,T,M)
```

```
630             CONTINUE
             ELSE
                READ(R,*,ERR=900,IOSTAT=HREAD) PRICET(N,T,1)
                DO 640 M=BEGINM,ENDM
                   PRICET(N,T,M) = PRICET(N,T,1)
640             CONTINUE
             ENDIF
620       CONTINUE
       ENDIF
600   CONTINUE


C  IF OPTION TO COMPUTE PRICE IS CHOSEN, CALL CORRESPONDING PRICE SUBR.
      DO 550 N=1,NOOFN
         K = K+1
         IF (INFO(K) .EQ. 0  .AND.  QOPN$(N) .EQ. 'Y') THEN
            WRITE(WI,52000) K, NAMEN(N)
            IF (N .EQ. 1)  CALL PRICE1(WRT2,RED2,HREAD,RUN2,RUN3)
            IF (N .EQ. 2)  CALL PRICE2(WRT2,RED2,HREAD,RUN2,RUN3)
            IF (N .EQ. 3)  CALL PRICE3(WRT2,RED2,HREAD,RUN2,RUN3)
            IF (N .EQ. 4)  CALL PRICE4(WRT2,RED2,HREAD,RUN2,RUN3)
            IF (N .EQ. 5)  CALL PRICE5(WRT2,RED2,HREAD,RUN2,RUN3)
            IF (N .EQ. 6)  CALL PRICE6(WRT2,RED2,HREAD,RUN2,RUN3)
            IF (HREAD .NE. 0)  GOTO 900
         ENDIF
550   CONTINUE


C  INPUT QUAN. IF OPTION TO COMPUTE QUANT. OF N IN STAGE,I, ISNT CHOSEN
      IF(RUN1 .AND. (OPTION .EQ. 1 .OR. OPTION .EQ. 2)) WRITE(W,41000)
      DO 700 N=1,NOOFN

C        DOES THE QUANTITY VARY BY MONTH?
         K = K+1
         IF (INFO(K) .EQ. 0  .AND.  ANYNOK(N) .NE. 0)  THEN
            IF (OPTION  .EQ.  1  .OR.  OPTION. EQ. 2)  THEN
               IF ((ENDM-BEGINM) .GT. 0) THEN
                  WRITE(W,40000) K, NAMEN(N)
                  READ(R,1000)  NQBYM(N)
C                 SET RELEVANT INFO LINES TO 0
                  INFO(K+1) = 0
               ELSE
                  NQBYM(N) = '-'
               ENDIF
            ENDIF
         ENDIF

C  QUANTITY OF EACH TYPE,T, OF EACH INPUT,N, USED IN EACH MONTH,M
         K = K+1
         IF (INFO(K) .EQ. 0) THEN
            IF (OPTION .EQ. 1  .OR.  OPTION .EQ. 2) THEN
               WRITE(W,41100) K, NAMEN(N), NAMENU(N)
               CHECK = .TRUE.
               DO 710 T=1,NOOFT(N)
                  IF (NOOFT(N) .GT. 1  .AND.  ANYNOK(N) .NE. 0)  THEN
                     WRITE (W,55000) NAMET(N,T)
                  ENDIF
                  DO 720 I=BEGINI,NOOFI
                    IF (OPNNEW(I,N) .EQ. 'E') THEN
                      WRITE(W,41300) I, NAMEI(I)
                      IF (NQBYM(N) .EQ. 'Y') THEN
                        DO 730 M=BEGINM,ENDM
                           WRITE(W,56000) NAMEM(M)
                           READ(R,*,ERR=900,IOSTAT=HREAD)TQUANI(I,N,T,M)
                           CHECK = .FALSE.
730                     CONTINUE
                      ELSE
                        READ(R,*,ERR=900,IOSTAT=HREAD)TQUANI(I,N,T,1)
                        DO 740 M=BEGINM,ENDM
                          TQUANI(I,N,T,M) = TQUANI(I,N,T,1)
740                     CONTINUE
                        CHECK = .FALSE.
                      ENDIF
                    ELSE
                      DO 750 M=BEGINM,ENDM
                        TQUANI(I,N,T,M) = -1.
750                   CONTINUE
                    ENDIF
720               CONTINUE
710            CONTINUE
               IF (CHECK)  WRITE(W,41500)
            ENDIF
         ENDIF
700   CONTINUE

      KLAST = K
      W    = WI
      R    = RI
      WRT2 = WI
      RED2 = RI


C  OPTION TO CHECK DATA
      WRITE(WI,60000)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y') THEN
         CALL ROUTS(WI)
      ENDIF

C  OPTION TO STOP OR REDO
      IF (RUN1) THEN
         WRITE(WI,64000)
         READ(RI,1000) QUEST2
         IF (QUEST2 .EQ. 'S')  THEN
            GOTO 850
         ELSEIF (QUEST2 .EQ. 'R') THEN
            GOTO 400
         ELSE
            RUN1 = .FALSE.
            RUN3 = .TRUE.
         ENDIF
      ENDIF

C  REASSIGN 1 TO INPUT LINES TO STOP FROM REENTERING DATA
      DO 800 K=1,NOOFK
         INFO(K) = 1
800   CONTINUE

C  OPTION TO CORRECT DATA, LINE BY LINE, BY CHOOSING A LINE
810   WRITE(WI,70000) NOOFK
      READ(RI,*) KGOTO
      IF (KGOTO .LT. 0  .OR.  KGOTO .GT. NOOFK) THEN
         GOTO 810
      ELSE IF (KGOTO .GT. 0) THEN
         INFO(KGOTO) = 0
         GOTO 400
      ENDIF
```

```
C  OPTION TO PRINT THE DATA ENTERED
850   WRITE(WI,80000)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y') THEN
         CALL ROUTS(WO)
      ENDIF

      IF(QUEST2.EQ.'S') STOP 'PROGRAM STOP IN DATA ENTRY SECTION 1'


1000  FORMAT (A1)

21000 FORMAT(' '/' ',I2,'. THE NUMBER OF THE MONTH THAT THE PRODUCTION',
     2              ' SEASON BEGINS'/' ',T6,'ENTER AN INTEGER:')

22000 FORMAT(' ',I2,'. THE NUMBER OF THE MONTH THAT THE PRODUCTION',
     2              ' SEASON ENDS' /' ',T6,'ENTER AN INTEGER:')

23000 FORMAT(' '/' ',I2,'. THE NUMBER OF DAYS IN EACH STAGE.')

23100 FORMAT(' ',T6,'IN THE',1X,A10,1X,'STAGE:')

24000 FORMAT(' '/' ',I2,'. THE NUMBER OF PERIODS IN EACH STAGE. ',
     2              'ENTER AN INTEGER:')

25000 FORMAT(' '//' ',I2,'. FOR OPTION 1, THE NUMBER OF DAYS UP TO THE'/
     2        ' ',T6,'FIRST STAGE OF INTEREST, ',A10)

26000 FORMAT(' ',I2,'. FOR OPTION 1, WHAT IS THE TOTAL COST UP TO THE'/
     2        ' ',4X,'FIRST STAGE OF INTEREST, ',A)

30000 FORMAT(' '//' ','YOU WILL NOW BE ASKED TO ENTER DATA FOR EACH OF',
     2                ' THE ',I2,' INPUTS'/)

31000 FORMAT(' '/' ',I2,'. THE NUMBER OF DISTINCT TYPES OF ',A/
     2        ' ',4X,'ENTER AN INTEGER GREATER OR EQUAL TO 1')

32000 FORMAT(' ',I2,'. THE NAME OF TYPE ',I2,' OF ',A/' ENTER UP TO ',
     2           '10 CHARACTERS, IN SINGLE QUOTES')

40000 FORMAT(' '/' ',I2,'. DOES THE QUANTITY OF ',A,' VARY BY MONTH?')

41000 FORMAT(' '//' ','QUANTITY INFORMATION:  (GENERAL SECTION)'/
     2             ' ','FOR OPTIONS 1 AND 2, ENTER THE QUANTITY OF ',
     3                'EACH INPUT USED EACH STAGE'/)

41100 FORMAT(' ',I2,'.',1X,'QUANTITY OF ',A,1X,A6,':')

41300 FORMAT(' ',5X,'IN STAGE',I2,', ',1X,A)

41500 FORMAT(' ',4X,'NO STAGES REQUIRE QUANTITIES OF THIS INPUT'/)

50000 FORMAT(' '/' ','FOR SOME INPUTS YOU CAN CHOOSE TO ',
     2       'LET THE PROGRAM COMPUTE INPUT PRICE OR'/
     3       ' ','YOU CAN CHOOSE TO INPUT THE PRICE/UNIT YOURSELF.')

50100 FORMAT(' '/' ','PRICE OF',1X,A,':')

51000 FORMAT(' ',I2,'. DO YOU WANT THE PROGRAM TO COMPUTE THE PRICE',
     2        ' ','OF ', A,'?')

51100 FORMAT(' ',I2,'. PRICE CAN NOT BE COMPUTED BY THE PROGRAM FOR ',A)

52000 FORMAT(' '//' ',I2,'. DATA FOR COMPUTING PRICE OF ',A,
     2        ' (PART OF GENERAL DATA SECTION)'/' ',5X,69('-'))

53000 FORMAT(' ',I2,'. DOES THE PRICE OF',1X,A,1X,'VARY BY MONTH?')

54000 FORMAT(' ',I2,'. PRICE OF',1X,A,1X,'PER',1X,A5)

55000 FORMAT(' ',4X,'OF TYPE',1X,A,':')

56000 FORMAT(' ',6X,'IN',1X,A)

60000 FORMAT(' '///' ','DO YOU WANT TO SEE THE GENERAL DATA?')

64000 FORMAT(' '/' ','ENTER  S  TO STOP EXECUTION, AFTER AN OPTION TO',
     2                ' PRINT GENERAL DATA.'/
     3        ' ','ENTER  R  TO REDO ALL DATA IN THIS SECTION ',
     3        ' INTERACTIVELY'/
     4        ' ','ENTER  C  TO CONTINUE EXECUTION WITH CORRECTIONS ',
     5              'LINE BY LINE'/
     6        ' ','ENTER S, R, OR C')

70000 FORMAT(' '/' YOU MAY CORRECT THE GENERAL DATA A LINE AT A TIME'/
     2        ' ',2X,'ENTER  0  TO MAKE NO CORRECTIONS OR'/
     3        ' ',2X,'ENTER THE NUMBER OF THE LINE YOU WANT TO CORRECT.'/
     4        ' ',2X,'(NOTE: IF YOU CHANGE MONTHS OF PRODUCTION, ALL ',
     5       'DATA MUST BE REINPUT.'/' ',2X,'CHANGES TO OTHER DATA ',
     6       'MAY FORCE REINPUT OF APPROPRIATE DATA.)'/
     7        ' ','ENTER AN INTEGER FROM 0 TO',1X,I2)

80000 FORMAT (' '/' ','DO YOU WANT A PRINTED COPY OF DATA ENTERED IN ',
     2             'THE GENERAL SECTION?')


900   RETURN
      END




      SUBROUTINE ROUTS(WRT)
C     ---------------------

C  ORGNSM:
C  ------
C  SUBROUTINE DOES NOT VARY
C  VARIABLES DO NOT CHANGE

C  STRUCTURE:  DISPLAYS OR PRINTS DATA FROM SUBROUTINE READS

C  1. PARAMETERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS
      INTEGER WRT
C  3. COMMON BLOCKS
      INCLUDE  (ICMAIN)
      INCLUDE  (ICBLK1)
      INCLUDE  (ICBLKO)
      INCLUDE  (ICUNIT)
C  4. LOCAL VARIABLES
      INTEGER  I, K, M, N, T, W

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------  ----------------------------------------------------
C  I       LOOP COUNTER FOR STAGE
```

```
C  K       INPUT LINE NUMBER
C  M       LOOP COUNTER FOR MONTHS
C  N       LOOP COUNTER FOR INPUTS
C  T       LOOP COUNTER FOR TYPES OF INPUT
C  W       UNIT NUMBER FOR WRITE
C  WRT     ARGUMENT FOR UNIT NUMBER TO WRITE
C  -------------------------------------------------------------


C  TRANSLATE ARGUMENTS TO LOCAL VARIABLES
      W = WRT

C  GENERAL INFORMATION
      WRITE(W,10000) NAME, DATE
      K = 1
         WRITE(W,11000) K, NAMEM(BEGINM), BEGINM
      K = K+1
         WRITE(W,12000) K, NAMEM(ENDM), ENDM
      WRITE(W,12500) (I, I=BEGINI,NOOFI)
      K = K+1
         WRITE(W,13000) K, (DAYSI(I), I=BEGINI,NOOFI)
      K = K+1
         WRITE(W,14000) K, (JPERI(I), I=BEGINI,NOOFI)
      K = K+1
         IF (OPTION .EQ. 1)  WRITE(W,15000) K, BEGINI, DAYOP1
      K = K+1
         IF (OPTION .EQ. 1)  WRITE(W,16000) K, BEGINI, CSOP1


C  CALCULATE OR INPUT PRICE?
      WRITE(W,40000)
      DO 500 N=1,NOOFN
         WRITE(W,41000) NAMEN(N)
         K = K+1
         WRITE(W,42000) K, QOPN$(N)
500   CONTINUE


C  LOOP THRU EACH N TO WRITE INFO WHEN PRICE OPTION WAS NOT CHOSEN
      DO 600 N=1,NOOFN
            WRITE(W,41000) NAMEN(N)

C           NUMBER AND NAME OF TYPES,T
            K = K+1
            IF (QOPN$(N) .EQ. 'N') THEN
               IF (OPTOK(N) .EQ. 0)  WRITE(W,31000) K, NOOFT(N)
            ENDIF
            K = K+1
            IF (QOPN$(N) .EQ. 'N') THEN
               IF (OPTOK(N) .EQ. 0  .AND.  NOOFT(N) .GT. 1)
     2         WRITE(W,32000)  K, (T, NAMET(N,T), T=1,NOOFT(N))
            ENDIF

C           DOES PRICE VARY BY MONTH?
            K = K+1
            IF (QOPN$(N) .EQ. 'N'  .AND. (ENDM-BEGINM) .GT. 0) THEN
               WRITE(W,44000)  K, N$BYM(N)
            ENDIF

C           PRICE OF EACH TYPE,T, IN EACH INPUT MONTH
            K = K+1
            IF (QOPN$(N) .EQ. 'N') THEN
             WRITE(W,45000)K,NAMEN(N),NAMENU(N),(NAMEM(M),M=BEGINM,ENDM)
            ELSE
             WRITE(W,45100)K,NAMEN(N),NAMENU(N),(NAMEM(M),M=BEGINM,ENDM)
            ENDIF
            DO 610 T=1,NOOFT(N)
               IF (NOOFT(N) .GT. 1)  WRITE(W,46000) NAMET(N,T)
               WRITE(W,46500) (PRICET(N,T,M), M=BEGINM,ENDM)
610         CONTINUE

600   CONTINUE


C  IF OPTION TO COMPUTE PRICE WAS CHOSEN, WRITE THE NO. OF EACH $ SUBR.
      DO 550 N=1,NOOFN
         K = K+1
         IF (QOPN$(N) .EQ. 'Y')  WRITE(W,43000) K, NAMEN(N)
550   CONTINUE

      IF (OPTION .EQ. 1  .OR.  OPTION .EQ. 2)  WRITE(W,33000)
       DO 700 N=1,NOOFN
          WRITE(W,34010)  NAMEN(N), NAMENU(N)

C        DOES THE QUANTITY VARY BY MONTH?
         K = K+1
         IF ((OPTION.EQ.1 .OR. OPTION.EQ.2) .AND.(ENDM-BEGINM).GT.0)THEN
            WRITE (W,34100) K, NQBYM(N)
         ENDIF

C        WHAT IS THE QUANTITY OF EACH INPUT USED?
         K = K+1
         IF (OPTION .EQ. 1  .OR.  OPTION .EQ. 2) THEN
            WRITE(W,34000) K, (I, I=BEGINI,NOOFI)
            DO 710 T=1,NOOFT(N)
               IF (NOOFT(N) .GT. 1)  WRITE(W,35000) NAMET(N,T)
               DO 720 M=BEGINM,ENDM
                 WRITE(W,36000)NAMEM(M),(TQUANI(I,N,T,M),I=BEGINI,NOOFI)
720            CONTINUE
710         CONTINUE
         ENDIF
700   CONTINUE


10000 FORMAT('1',A,' COST SIMULATION MODEL:  INPUT ENTERED (CONT.)'/
     2       ' ',A/' ','GENERAL DATA '/)

11000 FORMAT(' '/' ',I2,'. MONTH THAT PRODUCTION SEASON BEGINS:',T43,
     2              A10,'(',I2,')')

12000 FORMAT(' ',I2,'. MONTH THAT PRODUCTION SEASON ENDS:',T43,
     2          A10,'(',I2,')')

12500 FORMAT(' '//' ',T6,7(I2,' STAGE',2X)/' ',T6,7(8('-'),2X))

13000 FORMAT(' '/' ',I2,'. NUMBER OF DAYS IN EACH STAGE:'/
     2          ' ',T5,7(F9.2,1X))

14000 FORMAT(' '/' ',I2,'. NUMBER OF PERIODS IN EACH STAGE:'/
     2       ' ',3X,7(I8,2X))

15000 FORMAT(' '//' ',I2,'. FOR OPTION 1, THE NUMBER OF DAYS ',
     2         'UP TO STAGE ',I1,':',T54,F10.2)

16000 FORMAT(' ',I2,'. FOR OPTION 1, THE COST UP TO STAGE ',
     2              I1,':',T54,F10.2//)

31000 FORMAT(' ',I2,'. NUMBER OF TYPES :',I2)
```

```
32000 FORMAT(' ',I2,'. TYPE NAMES:'/' ',3X,4(5(I2,'=',A10,1X)/' '))

33000 FORMAT(' '//' ','QUANTITY USED OF EACH INPUT IN EACH STAGE  :'/
     2            ' ','(WHERE = -1, PROGRAM ',
     3            ' WILL COMPUTE THE QUANTITY LATER IN EXECUTION')

34010 FORMAT(' '/' ',A,1X,A,':')

34100 FORMAT(' ',I2,'. QUANTITY VARIES BY MONTH?:',2X,A)

34000 FORMAT(' ',I2,'. QUANTITY USED:'/' ',T21,7(2X,I2,'STAGE',6X))

35000 FORMAT(' ',4X,'OF',1X,A)

36000 FORMAT(' ',5X,'IN',1X,A10,':',T20,7(G14.7,1X))

40000 FORMAT(' ','INFORMATION PERTAINING TO PRICE'/' ',31(' '))

41000 FORMAT(' '/' ','INPUT:',2X,A)

42000 FORMAT(' ',I2,'. PRICE COMPUTED BY PROGRAM?:',1X,A)

43000 FORMAT(' '/' ',I2,'. SECTION TO COMPUTE',1X,A,1X,'PRICE  ',
     2       '(SEE SEPARATE INPUT LIST)')

44000 FORMAT(' ',I2,'. PRICE VARIES BY MONTH?:',2X,A)

45000 FORMAT(' ',I2,'. PRICE OF ',A, ' PER ' ,A,':'/' ',T18,6(A,5X)/
     2       ' ',T18,6(A,5X))

45100 FORMAT(' ',I2,'. PRICE OF ',A,' PER ',A,' (COMPUTED BY THE ',
     2       'PROGRAM):'/' ',T18,6(A10,5X)/' ',T18,6(A10,5X))

46000 FORMAT(' ',T6,A10,':')

46500 FORMAT(' ',T16,6(G14.7,1X)/' ',T16,6(G14.7,1X))


      RETURN
      END







      SUBROUTINE READO(WRT,RED,HREAD,RUN2)
C     -----------------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE WITH ORGANISM
C  VARIABLES THAT MUST BE RETURNED:
C    THE TYPE OF THE ORGANISM          :  NAMEO(O)
C    THE NO. OF ORGANISMS AT BEGINNING :  ONOBEG
C    THE TYPE OF SYSTEM USED           :  NAMES
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE: READS IN THE ABOVE MENTIONED DATA
C             READS IN OTHER DATA SPECIFIC TO ORGANISM
C             OPTION TO CHECK, CORRECT, OR PRINT DATA
```

```

C  1. PARAMETERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS
      INTEGER  HREAD, RED, RED2, WRT, WRT2, WW
      LOGICAL  RUN2
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICBLK1)
      INCLUDE (ICBLKO)
      INCLUDE (ICUNIT)
      INCLUDE (ICORG)
C  4. LOCAL VARIABLES
      INTEGER  O, INFO, K, KGOTO, KLAST, NOOFK, R, W
      LOGICAL   RUN1
      CHARACTER * 1  QUEST, QUEST2
      PARAMETER  (NOOFK = 8)
      DIMENSION  INFO(NOOFK)

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------  ----------------------------------------------------
C  O       LOOP COUNTER FOR ORGANISM
C  HREAD   ARGUE:  ISOSTAT TO INDICATE PROBLEMS IN READING FILE
C  INFO    INFORMATION LINE NUMBER
C  K       NUMBER OF THE INFO LINE
C  KGOTO   LINE NUMBER TO CORRECT
C  KLAST   LAST INFO LINE NUMBER TO BE INPUT
C  NOOFK   NUMBER OF INPUT LINES
C  QUEST   GENERAL VARIABLE FOR USER QUESTION (IS WRITTEN OVER)
C  QUEST2  OPTION TO STOP, REDO, OR CONTINUE DATA ENTRY
C  R       UNIT NUMBER FOR READ
C  RED     ARGUE:  FOR UNIT NUMBERS TO READ
C  RED2    ARGUE:  UNIT NO. TO READ FOR NEXT LEVEL SUBR.
C  RUN1    FIRST RUN THROUGH
C  RUN2    ARGUE:  INDICATES IF ISNT FIRST RUN THROUGH
C  W       UNIT NUMBER FOR WRITE
C  WRT     ARGUE:  FOR UNIT NUMBERS TO WRITE
C  WRT2    ARGUE:  UNIT NO. TO WRITE FOR NEXT LEVEL SUBR.
C  WW      ARGUE:  UNIT NO. TO WRITE TO OUTPUT FILE
C  ------------------------------------------------------------------


C  NOT FIRST RUN, IMMEDIATLY OFFER OPTION TO CHECK AND CHANGE DATA
      IF (RUN2) THEN
         DO 120  K=1,NOOFK
            INFO(K) = 1
120      CONTINUE
         WRT2 = WI
         RED2 = RI
         W    = WI
         R    = RI
         WRITE(WI,60000) NAME
         RUN1 = .FALSE.
         READ(RI,1000)  QUEST
         IF (QUEST .EQ. 'Y') CALL ROUTO(WI)

100      WRITE(WI,70000) NAME, NOOFK
         READ(RI,*) KGOTO
         IF (KGOTO .LT. 0  .OR.  KGOTO .GT. NOOFK) THEN
            GOTO 100
         ELSEIF (KGOTO .GT. 0) THEN
            INFO (KGOTO) = 0
         ELSEIF (KGOTO .EQ. 0)  THEN
            GOTO 850
```

```
        ENDIF

      ELSE
C     INITIALIZE INPUT LINES TO READ IN ALL DATA THE FIRST RUN
C     ASSIGN ARGUE. TO LOCAL UNIT NUMBERS TO BE CONSISTENT W/ OPTION
        W = WRT
        R = RED
        WRT2 = WRT
        RED2 = RED
        DO 110 K=1,NOOFK
           INFO(K) = 0
110     CONTINUE
        RUN1 = .TRUE.
      ENDIF


C  READ IN DATA SPECIFIC TO ORGANISM
400   HREAD = 0

C  THE TYPE OF ORGANISM
      K = 1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,20000) K, NAME
         DO 405 O=1,NOOFO
            WRITE(W,20100) O, NAMEO(O)
405      CONTINUE
         READ(R,*,ERR=900,IOSTAT=HREAD) OTYPE
      ENDIF

C  THE TYPE OF SYSTEM USED
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,20015) K
         READ (R,*,ERR=900,IOSTAT=HREAD) NAMES
      ENDIF

C  INFORMATION CONCERNING CONDITIONING STAGE
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,20020) K, NAMEI(1)
         READ (R,*,ERR=900,IOSTAT=HREAD) DAYSI1
      ENDIF
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,20025) K, NAMEO(OTYPE), NAMEI(1)
         READ (R,*,ERR=900,IOSTAT=HREAD) NOOI1
      ENDIF

C  NUMBER OF ORGANISMS THAT ARE BEGUN
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,30000) K, NAMEO(OTYPE), NAMEI(2)
         READ(R,*,ERR=900,IOSTAT=HREAD) ONOBEG
      ENDIF

C  CHOOSE PARAMETERS FOR AGE NUMBER ALGORITHM
      IF (RUN1)  WRITE(W,40000)  EQNO(1)
      K = K+1
      IF (INFO(K) .EQ. 0)  THEN
         WRITE (W,41000)  K, EQNOAF
         READ (R,*,ERR=900,IOSTAT=HREAD)  EQNOA
      ENDIF

C  CHOOSE PARAMETERS FOR SIZE/AGE ALGORITHM
      IF (RUN1)  WRITE(W,50000)  EQSZE(1)
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,51000)  K, EQSZAF
         READ(R,*,ERR=900,IOSTAT=HREAD)  EQSZA
      ENDIF
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,52000)  K, EQSZBF
         READ(R,*,ERR=900,IOSTAT=HREAD)  EQSZB
      ENDIF


      KLAST = K
      WRT2 = WI
      RED2 = RI
      W    = WI
      R    = RI


C  OPTION TO CHECK DATA
      WRITE(WI,60000) NAME
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y') THEN
         CALL ROUTO(WI)
      ENDIF

C  OPTION TO STOP OR REDO
      IF (RUN1) THEN
         WRITE(WI,64000) NAME
         READ(RI,1000) QUEST2
         IF (QUEST2 .EQ. 'S')  THEN
            GOTO 850
         ELSEIF (QUEST2 .EQ. 'R')  THEN
            GOTO 400
         ENDIF
         RUN1 = .FALSE.
      ENDIF

C  REASSIGN 1 TO INPUT LINES TO STOP FROM REENTERING DATA
      DO 800 K=1,NOOFK
         INFO(K) = 1
800   CONTINUE

C  OPTION TO CORRECT DATA, LINE BY LINE, BY CHOOSING A LINE
810   WRITE(WI,70000) NAME, NOOFK
      READ(RI,*) KGOTO
      IF (KGOTO .LT. 0  .OR.  KGOTO .GT. NOOFK) THEN
         GOTO 810
      ELSE IF (KGOTO .GT. 0) THEN
         INFO(KGOTO) = 0
         GOTO 400
      ENDIF

C  OPTION TO PRINT THE DATA ENTERED
850   WRITE(WI,80000 ) NAME
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y') THEN
         CALL ROUTO(WO)
      ENDIF

      IF(QUEST2.EQ.'S') STOP 'PROGRAM STOP IN DATA ENTRY SECTION 1'
```

```
1000  FORMAT (A1)

20000 FORMAT(' ',I2,'. CHOOSE ONE OF THE FOLLOWING ',A,' TYPES:')

20015 FORMAT(' ',I2,'. NAME OF THE SYSTEM USED.'/' ',4X,'UP TO 50  ',
     2             'CHARACTERS IN QUOTES')

20100 FORMAT(' ',T6,I1,' = ',A)

20020 FORMAT(' ',I2,'. NUMBER OF DAYS IN THE ',A,' STAGE.')

20025 FORMAT(' ',I2,'. NUMBER OF ',A,' IN THE ',A,' STAGE')

30000 FORMAT(' ',I2,'. NUMBER OF ',A,' THAT A BATCH IS BEGUN WITH IN ',
     2       'THE ',A,' STAGE')

40000 FORMAT(' '/' THE FOLLOWING EQUATION IS USED TO EQUATE AGE IN ',
     2           'WEEKS TO NUMBER SURVIVING:'/' ',4X,A)

41000 FORMAT(' ',I2,'. CHOOSE SURVIVAL PARAMETER A (SUGGESTED: ',
     2                 G14.7,')')

50000 FORMAT(' '/' THE FOLLOWING EQUATION IS USED TO EQUATE AGE IN ',
     2            'WEEKS TO SIZE IN MM:'/' ',4X,A)

51000 FORMAT(' ',I2,'. CHOOSE SIZE PARAMETER A (SUGGESTED:',
     2           G14.7,')')

52000 FORMAT(' ',I2,'. CHOOSE SIZE PARAMETER B (SUGGESTED: ',
     2           G14.7,')')

60000 FORMAT(' '/' ','DO YOU WANT TO CHECK DATA FOR ',A/)

64000 FORMAT(' ','ENTER  S  TO STOP EXECUTION, AFTER AN OPTION TO',
     2                ' PRINT DATA FOR ',A/
     3        ' ','ENTER  R  TO REDO ALL DATA IN THIS SECTION ',
     3        'INTERACTIVELY'/
     4        ' ','ENTER  C  TO CONTINUE EXECUTION WITH CORRECTIONS ',
     5              'LINE BY LINE'/
     6        ' ','ENTER S, R, OR C'/)

70000 FORMAT(' ','YOU MAY CORRECT DATA LINE AT A TIME FOR ',A/
     2        ' ',2X,'ENTER  0  TO MAKE NO CORRECTIONS OR'/
     3        ' ',2X,'ENTER THE NUMBER OF THE LINE YOU WANT TO CORRECT.'/
     4        ' ','ENTER AN INTEGER FROM 0 TO',1X,I2)

80000 FORMAT (' '/' DO YOU WANT A PRINTED COPY OF DATA ENTERED FOR ',A,
     2        ' SECTION?')


900   RETURN
      END





      SUBROUTINE ROUTO(WRT)
C     ---------------------

C ORGNSM:
C ------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE WITH ORGANISM, BUT SOME MUST BE RETURNED
C  VARIABLES REQUIRED ARE THOSE REQUIRED IN READO
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  DISPLAYS OR PRINTS DATA FROM SUBROUTINE READO

C  1. PARAMETERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS
      INTEGER WRT
C  3. COMMON BLOCKS
      INCLUDE  (ICMAIN)
      INCLUDE  (ICBLK1)
      INCLUDE  (ICBLKO)
      INCLUDE  (ICUNIT)
      INCLUDE  (ICORG)
C  4. LOCAL VARIABLES
      INTEGER  K, W

C NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C ------  -----------------------------------------------------
C K       INPUT LINE NUMBER
C W       UNIT NUMBER FOR WRITE
C WRT     ARGUE:  UNIT NUMBER FOR WRITE
C ---------------------------------------------------------------


C TRANSLATE ARGUMENTS TO LOCAL VARIABLES
     W = WRT

C GENERAL INFORMATION
     WRITE(W,10000) NAME, DATE, NAME
     K = 1
        WRITE(W,10500) K, NAME, OTYPE, NAMEO(OTYPE)
     K = K+ 1
        WRITE(W,10515) K, NAMES
     K = K+1
        WRITE(W,10520) K, NAMEI(1), DAYSI1
     K = K+1
        WRITE(W,10525) K, NAMEI(1), NOOI1
     K = K+1
        WRITE(W,30000) K, NAMEO(OTYPE), ONOBEG

C PARAMETERS FOR SIZE/NUMBER EQUATION
     WRITE(W,40000)  EQNO(1)
     K = K+1
        WRITE(W,41000)  K, EQNOA

C PARAMETERS FOR AGE/SIZE EQUATION
     WRITE(W,50000)  EQSZE(1)
     K = K+1
        WRITE(W,51000)  K, EQSZA
     K = K+1
        WRITE(W,52000)  K, EQSZB


10000 FORMAT('1',A,' COST SIMULATION MODEL:  INPUT ENTERED (CONT.)'/
     2        ' ',A/' ',A,' DATA '/)

10500 FORMAT(' ',I2,'. TYPE OF ',A,T46,I2,' =',A10)

10515 FORMAT(' '/' ',I2,'. NAME OF SYSTEM',T46,A)
```

```
10520 FORMAT(' '/' ',I2,'. NUMBER DAYS IN STAGE ',A,T46,G10.5)

10525 FORMAT(' '/' ',I2,'. NUMBER BIVALVES IN STAGE ',A,T46,G10.5)

30000 FORMAT(' '/' ',I2,'. NUMBER OF ',A,' BATCH BEGUN WITH:',T46,G12.7)

40000 FORMAT(' '/' ','EQUATION TO DETERMINE SURVIVAL:'/' ',4X,A)

41000 FORMAT(' ',I2,'. PARAMETER A = ',G14.7)

50000 FORMAT(' '/' ','EQUATION TO EQUATE AGE TO SIZE:'/' ',4X,A)

51000 FORMAT(' ',I2,'. PARAMETER A = ',G14.7)

52000 FORMAT(' ',I2,'. PARAMETER B = ',G14.7/)


      RETURN
      END







      SUBROUTINE READN1(WRT,RED,HREAD,RUN2)
      RETURN
      END




      SUBROUTINE READN2(WRT,RED,HREAD,RUN2)
      RETURN
      END




      SUBROUTINE READN3(WRT,RED,HREAD,RUN2)
      RETURN
      END




      SUBROUTINE READN4(WRT,RED,HREAD,RUN2)
      RETURN
      END




      SUBROUTINE READN5(WRT,RED,HREAD,RUN2)
C     -------------------------------------

C ORGNSM:
C ------
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE WITH ORGANISM
C VARIABLES NOT RETURNED TO ORGNSM
C VARIABLES NECESSARY ONLY FOR QUANT5, IF OPTION TO COMPUTE QUANTITY=Y
C CURRENT ORGANISM:  BIVALVES

C STRUCTURE:   READS IN DATA NECESSARY TO COMPUTE QUANTITY OF INPUT 5
C              OPTIONS TO CHECK, PRINT, CORRECT DATA

C 1. PARAMETERS
     INCLUDE  (IPAR1)
C 2. ARGUMENTS
     INTEGER  HREAD, RED, RED2, WRT, WRT2
     LOGICAL  RUN2
C 3. COMMON BLOCKS
     INCLUDE (ICMAIN)
     INCLUDE (ICBLK1)
     INCLUDE (ICBLKO)
     INCLUDE (ICUNIT)
     INCLUDE (ICORG)
C 4. LOCAL VARIABLES
     INTEGER  EQ, I, INFO, K, KGOTO, KLAST, M, NOOFK, R, T, W
     LOGICAL  RUN1
     CHARACTER * 1  QUEST, QUEST2
     PARAMETER  (NOOFK=9)
     DIMENSION  INFO(NOOFK)

C NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C ------  -------------------------------------------------------
C EQ      LOOP COUNTER FOR EQUATIONS
C HREAD   ARGUE:  IOSTAT TO INDICATE PROBLEMS READING FILE
C I       LOOP COUNTER FOR INPUTS
C INFO    INDICATOR OF INPUT LINE
C K       INPUT LINE NUMBER
C KGOTO   INPUT LINE NO. TO GOTO TO CORRECT
C KLAST   LAST INPUT LINE INPUT
C M       LOOP COUNTER FOR MONTHS
C NOOFK   NUMBER OF INPUT LINES
C QUEST   GENERAL INTERACTIVE USER QUESTION (IS OVERWRITTEN)
C QUEST2  OPTION TO STOP, REDO, OR CONTINUE
C R       UNIT NUMBER FOR READ
C RED     ARGUE:  UNIT NUMBER FOR READ
C RED2    ARGUE:  UNIT NUMBER FOR READ FOR NEXT LEVEL SUBR.
C RUN1    IS IT THE FIRST RUN?
C RUN2    ARGUE: IS IT GREATER THAN THE FIRST RUN
C T       LOOP COUNTER FOR TYPES
C W       UNIT NUMBER FOR WRITE
C WRT     ARGUE:  UNIT NUMBER FOR WRITE
C WRT2    ARGUE:  UNIT NUMBER TO WRITE TO NEXT LEVEL SUBROUTINE
C ---------------------------------------------------------------


C IF NOT 1ST RUN, IMMEDIATELY OFFER TO DISPLAY AND CHANGE DATA
     IF (RUN2) THEN
        DO 120  K=1,NOOFK
           INFO(K) = 1
120     CONTINUE
        WRT2 = WI
        RED2 = RI
        W    = WI
        R    = RI
        RUN1 = .FALSE.
        WRITE(WI,60000)  NAMEN(5)
        READ(RI,1000) QUEST
        IF (QUEST .EQ. 'Y')  CALL ROUTN5(WI)

100     WRITE(WI,70000) NAMEN(5), NOOFK
        READ(RI,*) KGOTO
        IF (KGOTO .LT. 0  .OR.  KGOTO .GT. NOOFK) THEN
           GOTO 100
```

```
         ELSEIF (KGOTO .GT. 0)  THEN
            INFO(KGOTO) = 0
         ELSEIF (KGOTO .EQ. 0)  THEN
            GOTO 850
         ENDIF

      ELSE
C     INITIALIZE VARIABLES TO 0 TO READ IN DATA FIRST TIME
C     ASSIGN ARGUE UNIT NO. TO LOCAL UNIT NUMBERS
         DO 110 K=1,NOOFK
            INFO(K) = 0
110      CONTINUE
         RUN1 = .TRUE.
         W = WRT
         R = RED
         WRT2 = WRT
         RED2 = RED
      ENDIF


C  READ IN ALL DATA
400   HREAD = 0

C  CHOOSE ALGORITHM AND PARAMETERS TO COMPUTE QUANTITY OF ALGAE
      K = 1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,10000)  K
         WRITE(W,10100)  (EQ, EQQ5(EQ), EQ=1,2)
         READ(R,*,ERR=900,IOSTAT=HREAD)  QEQQ5
         INFO(K+1) = 0
         INFO(K+2) = 0
      ENDIF

      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,11000)  K, EQQ5AF(QEQQ5)
         READ(R,*,ERR=900,IOSTAT=HREAD)  EQQ5A(QEQQ5)
      ENDIF
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,11100)  K, EQQ5BF(QEQQ5)
         READ(R,*,ERR=900,IOSTAT=HREAD)  EQQ5B(QEQQ5)
      ENDIF

C  FACTORS AFFECTING QUANTITY REQUIRED
C  FEEDING FILTERING RATE
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         WRITE(W,20000) K
         DO 200 I=BEGINI,NOOFI
            IF (OPNNEW(I,5) .EQ. 'C') THEN
               WRITE(W,20100)  I, NAMEI(I)
               READ(R,*,ERR=900,IOSTAT=HREAD)  FILTER(I)
            ENDIF
200      CONTINUE
      ENDIF

C  FRACTION OF EACH TYPE OF ALGAE (IF ¢ 1) USED IN EACH STAGE
      K = K+1
      IF (INFO(K) .EQ. 0)  THEN
         IF (NOOFT(5) .GT. 1) THEN
            WRITE(W,21000)  K
            DO 210  I=BEGINI,NOOFI
               IF (OPNNEW(I,5) .EQ. 'C') THEN
                  WRITE(W,20100)  I, NAMEI(I)
                  DO 211  T=1,NOOFT(5)
                     IF (NOOFT(5) .GT. 1)  WRITE(W,20200) NAMET(5,T)
                     READ(R,*,ERR=900,IOSTAT=HREAD)  N5REQ(I,T)
211               CONTINUE
               ENDIF
210         CONTINUE
         ELSE
            DO 220 I=BEGINI,NOOFI
               IF (OPNNEW(I,5) .EQ. 'C') THEN
                  N5REQ(I,1) = 1.
               ENDIF
220         CONTINUE
         ENDIF
      ENDIF

C  AVAILABILITY OF ALGAE BY MONTH FROM SEAWATER, PRODUCED AND PURCHASED
      K = K+1
      IF (INFO(K) .EQ. 0)  THEN
         IF ( (ENDM-BEGINM) .GT. 0)  THEN
            WRITE(W,30000)  K
            READ(R,1000,ERR=900,IOSTAT=HREAD)  SEABYM
         ELSE
            SEABYM = '-'
         ENDIF
         INFO(K+1) = 0
      ENDIF
      K = K+1
      IF (INFO(K) .EQ. 0)  THEN
         WRITE(W,30100)  K
         DO 300  I=BEGINI,NOOFI
            IF (OPNNEW(I,5) .EQ. 'C') THEN
               WRITE(W,20100)  I, NAMEI(I)
               IF (SEABYM .EQ. 'Y')  THEN
                  DO 301  M=BEGINM,ENDM
                     WRITE(W,30200)  NAMEM(M)
                     READ(R,*,ERR=900,IOSTAT=HREAD)  SEACEL(I,M)
301               CONTINUE
               ELSE
                  READ(R,*,ERR=900,IOSTAT=HREAD)  SEACEL(I,1)
                  DO 302  M=BEGINM,ENDM
                     SEACEL(I,M) = SEACEL(I,1)
302               CONTINUE
               ENDIF
            ENDIF
300      CONTINUE
      ENDIF

C  QUANTITY OF ALGAE PRODUCED IF PRICE OPTION NOT CHOSEN
C  IF PRICE OPTION CHOSEN, THEN PRICE SUBR. SHOULD DETERMINE QUANT.
      K = K+1
      IF (INFO(K) .EQ. 0) THEN
         IF (QOPN$(5) .EQ. 'N') THEN
            IF ( (ENDM-BEGINM) .GT. 0) THEN
               WRITE(W,31000)  K
               READ(R,1000,ERR=900,IOSTAT=HREAD)  N5ABYM
            ELSE
               N5ABYM = '-'
            ENDIF
            INFO(K+1) = 0
         ENDIF
      ENDIF
      K = K+1
      IF (INFO(K) .EQ. 0)  THEN
```

```
      IF (QOPN$(5) .EQ. 'N') THEN
         WRITE(W,31100) K
         DO 311  T=1,NOOFT(5)
            WRITE(W,20200)  NAMET(5,T)
            IF (N5ABYM .EQ. 'Y')  THEN
               DO 312  M=BEGINM,ENDM
                  WRITE(W,30200)  NAMEM(M)
                  READ(R,*,ERR=900,IOSTAT=HREAD) N5AVAL(T,M)
312            CONTINUE
            ELSE
               READ(R,*,ERR=900,IOSTAT=HREAD)  N5AVAL(T,1)
               DO 313  M=BEGINM,ENDM
                  N5AVAL(T,M) = N5AVAL(T,1)
313            CONTINUE
            ENDIF
311      CONTINUE
      ENDIF
   ENDIF


   KLAST = K
   WRT2 = WI
   RED2 = RI
   W    = WI
   R    = RI


C  OPTION TO CHECK DATA
   WRITE(WI,60000)  NAMEN(5)
   READ(RI,1000) QUEST
   IF (QUEST .EQ. 'Y') THEN
      CALL ROUTN5(WI)
   ENDIF

C  OPTION TO STOP OR REDO
   IF (RUN1) THEN
      WRITE(WI,64000) NAMEN(5)
      READ(RI,1000) QUEST2
      IF (QUEST2 .EQ. 'S')  THEN
         GOTO 850
      ELSEIF (QUEST2 .EQ. 'R') THEN
         GOTO 400
      ELSEIF (QUEST .EQ. 'C') THEN
         CONTINUE
      ENDIF
      RUN1 = .FALSE.
   ENDIF

C  REASSIGN 1 TO INPUT LINES TO STOP FROM REENTERING DATA
   DO 800 K=1,NOOFK
      INFO(K) = 1
800   CONTINUE

C  OPTION TO CORRECT DATA, LINE BY LINE, BY CHOOSING A LINE
810   WRITE(WI,70000) NAMEN(5), NOOFK
   READ(RI,*) KGOTO
   IF (KGOTO .LT. 0  .OR.  KGOTO .GT. NOOFK) THEN
      GOTO 810
   ELSEIF (KGOTO .GT. 0) THEN
      INFO(KGOTO) = 0
      GOTO 400
   ENDIF

C  OPTION TO PRINT THE DATA ENTERED
850   WRITE(WI,80000) NAMEN(5)
   READ(RI,1000) QUEST
   IF (QUEST .EQ. 'Y') THEN
      CALL ROUTN5(WO)
   ENDIF

   IF(QUEST2.EQ.'S') STOP 'PROGRAM STOP IN DATA ENTRY INPUT 5'

C  OPTION TO PRINT OUT ADDITIONAL INFORMATION ABOUT INPUT 5
   IF (OUTQN(5) .EQ. 0) THEN
      WRITE(WI,90000)
      READ(RI,1000)  QOUTQN(5)
      IF (QOUTQN(5) .EQ. 'Y') THEN
         WRITE(WI,91000)  WO
         READ(RI,*) WON(5)
      ENDIF
   ENDIF


1000  FORMAT (A1)

10000 FORMAT(' '/' ',I2,'. TO COMPUTE ALGAE QUANTITY, CHOOSE ONE OF ',
    2          'THE FOLLOWING EQUATIONS'/' ',4X,'THAT BEST ',
    2        'RELATES AGE IN WEEKS TO 10,000 CELLS CONSUMED.'/
    3     ' ',4X,'(THE EQUATION WILL BE SOLVED FOR THE NUMBER ',
    4        'OF CELLS CONSUMED PER DAY.)'/)

10100 FORMAT(' ',3X,I2,':  ',A)

11000 FORMAT(' ',I2,'. CHOOSE ALGAE PARAMETER A (SUGGESTED: ',
    2        G14.7,')')

11100 FORMAT(' ',I2,'. CHOOSE ALGAE PARAMETER B (SUGGESTED: ',
    2        G14.7,')')

20000 FORMAT(' ',I2,'. WHAT IS THE FEEDING FILTERING RATE OF THE ',
    2     'ORGANISM AT EACH STAGE'/
    3     ' ',4X,'FOR WHICH ALGAE QUANTITY IS TO BE COMPUTED')

20100 FORMAT(' ',3X,' IN STAGE ',I2,', ',A)

21000 FORMAT('0',I2,'. WHAT IS THE FRACTION OF EACH TYPE OF ALGAE ',
    2        'REQUIRED IN EACH STAGE (TOTAL 100%):')

20200 FORMAT(' ',6X,'OF TYPE ',A)

30000 FORMAT(' '/' ',I2,'. DOES THE AVAILABILITY OF ALGAE FROM ',
    2        'SEAWATER CHANGE BY MONTH?')

30100 FORMAT(' ',I2,'. FRACTION OF DIET BIVALVES RECEIVE FROM ',
    2     'SEAWATER:')

30200 FORMAT(' ',7X,A)

31000 FORMAT(' '/' ',I2,'. DOES THE AVAILABILITY OF ALGAE PRODUCED ',
    2     'CHANGE BY MONTH?')

31100 FORMAT(' ',I2,'. TOTAL AMOUNT OF EACH TYPE OF ALGAE AVAILABLE')

60000 FORMAT(' '/' ','DO YOU WANT TO CHECK THE DATA ENTERED FOR ',A)

64000 FORMAT(' '/' ','ENTER  S  TO STOP EXECUTION AFTER AN OPTION TO ',
    2          'PRINT DATA FOR ',A/
    3     ' ','ENTER  R  TO REDO ALL ALGAE DATA INTERACTIVELY'/
```

```
     4        ' ','ENTER  C  TO CONTINUE WITH OPTION TO CORRECT ',
     5           'LINE BY LINE'/
     6        ' ','ENTER S, R, OR C')

70000 FORMAT (' '/' YOU MAY NOW CORRECT DATA A LINE AT A TIME FOR ',A/
     2         ' ',2X,'ENTER 0 TO MAKE NO CORRECTIONS, OR'/
     3         ' ',2X,'ENTER THE NO. OF THE LINE YOU WANT TO CORRECT'/
     4         ' ','ENTER AN INTEGER FORM 0 TO',1X,I2)

80000 FORMAT (' '/' DO YOU WANT A PRINTED COPY OF DATA ENTERED FOR ',A,
     2         '?  Y OR N')

90000 FORMAT(' '/' ','DO YOU WANT TO PRINT ADDITIONAL ',
     2                'INFORMATION ABOUT ALGAE?  Y OR N')

91000 FORMAT(' ','TO WHAT OUTPUT FILE UNIT DO YOU WANT THIS TO GO TO?',
     2        ' - INTEGER 10 TO 99.'/' (MAIN OUTPUT FILE HAS ',
     3        'A UNIT = ',I2,')')


900   RETURN
      END




      SUBROUTINE READN6(WRT,RED,HREAD,RUN2)
      RETURN
      END







      SUBROUTINE ROUTN1(WRT)
      RETURN
      END




      SUBROUTINE ROUTN2(WRT)
      RETURN
      END



      SUBROUTINE ROUTN3(WRT)
      RETURN
      END




      SUBROUTINE ROUTN4(WRT)
      RETURN
      END




      SUBROUTINE ROUTN5(WRT)
C     ----------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE, BUT NOT RETURNED TO ORGNSM
C  VARIABLES REQUIRED ARE THOSE TO PRINT OUT VARIABLES ENTERED
C  IN READN5, WHICH ARE SPECIFIC TO THE ORGANISM
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE: DISPLAYS OR PRINTS OUT DATA FOR SUBROUTINE READN5

C  1. PARAMETERS
      INCLUDE(IPAR1)
C  2. ARGUMENTS
      INTEGER WRT
C  3. COMMON BLOCKS
      INCLUDE(ICBLK1)
      INCLUDE(ICBLKO)
      INCLUDE(ICMAIN)
      INCLUDE(ICORG)
C  4. LOCAL VARIABLES
      INTEGER  I, K, M, T, W

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------  ------------------------------------------------------
C  I       LOOP COUNTER FOR STAGE
C  K       INPUT LINE NUMBER
C  M       LOOP COUNTER FOR MONTH
C  T       LOOP COUNTER FOR TYPES
C  W       UNIT NUMBER FOR WRITE
C  WRT     ARGUMENT FOR UNIT NUMBER TO WRITE
C  ------------------------------------------------------------------


C  TRANSLATE ARGUMENTS TO LOCAL VARIABLES
      W = WRT

C  WRITE OUT HEADING
      WRITE(W,5000) NAME, DATE, NAMEN(5)

C  ALGORITHM TO COMPUTE QUANTITY
      K = 1
         WRITE(W,10000)  K, QEQQ5, EQQ5(QEQQ5)
      K = K+1
         WRITE(W,11000)  K, EQQ5A(QEQQ5)
      K = K+1
         WRITE(W,11100)  K, EQQ5B(QEQQ5)

      WRITE(W,20000) (I, I=BEGINI,NOOFI)

C  QUANTITY REQUIRED
C  FEEDING FILTERING RATE
      K = K + 1
         WRITE(W,21000)  K, (FILTER(I), I=BEGINI,NOOFI)
C  FRACTION OF EACH TYPE USED IN DIET
      K = K+1
      IF (NOOFT(5) .GT. 1)  THEN
         WRITE(W,21100) K
         DO 211 T=1,NOOFT(5)
            IF (NOOFT(5) .GT. 1)  WRITE(W,21200) NAMET(5,T)
            WRITE(W,21300) (N5REQ(I,T), I=BEGINI,NOOFI)
211      CONTINUE
      ENDIF
```

```
C  AVAILABILITY
      K = K+1
         IF ((ENDM-BEGINM) .GT. 0)  WRITE(W,30000) K, SEABYM
      K = K+1
         WRITE(W,30100) K
         DO 300 M=BEGINM,ENDM
            WRITE(W,30200) NAMEM(M), (SEACEL(I,M), I=BEGINI,NOOFI)
300      CONTINUE

      K = K+1
         IF (QOPN$(5) .EQ. 'N'  .AND.  (ENDM-BEGINM) .GT. 0)  THEN
            WRITE(W,31000)  K, N5ABYM
         ENDIF
      K = K+1
      IF (QOPN$(5) .EQ. 'N')  THEN
         WRITE(W,31100) K, (NAMEM(M), M=BEGINM,ENDM)
         DO 310 T=1,NOOFT(5)
            IF (NOOFT(5) .GT. 1)  WRITE(W,21200)  NAMET(5,T)
            WRITE(W,31300) (N5AVAL(T,M), M=BEGINM,ENDM)
310      CONTINUE
      ENDIF



5000  FORMAT('1',A,' COST SIMULATION MODEL:  INPUT ENTERED (CONT.)'/
     2         ' ',A50/' ',A,1X,'DATA' /)

10000 FORMAT(' ',I2,'. AGE/ALGAE EQUATION CHOSEN: ',I1/' ',4X,A)

11000 FORMAT(' ',I2,'. PARAMETER A = ',G14.7)

11100 FORMAT(' ',I2,'. PARAMETER B = ',G14.7/)

20000 FORMAT(' ','IN THE FOLLOWING, WHERE QUANTITY=0, QUANTITY IS ',
     2        'ENTERED'//' ',6X,6(I2,' STAGE',2X)/)

21000 FORMAT(' ',I2,'. FILTERING RATE'/' ',6X,6(G9.2,1X))

21100 FORMAT(' '/' ',I2,'. FRACTION OF EACH ALGAE TYPE REQUIRED  ',
     2        'PER STAGE:')

21200 FORMAT(' ',4X,' TYPE ',A)

21300 FORMAT(' ',6X,6(G9.2,1X))

30000 FORMAT(' '/' ',I2,'. FRACTION FROM SEA VARIES BY MONTH?: ',A1)

30100 FORMAT(' ',I2,'. FRACTION OF DIET FROM SEA ALGAE IN EACH ',
     2        'STAGE, EACH MONTH:')

30200 FORMAT(' ',6X,A/' ',6X,6(G9.2,1X)/)

31000 FORMAT(' '/' ',I2,'. AVAILABILITY OF ALGAE VARIES BY MONTH?: ',A)

31100 FORMAT(' ',I2,'. AVAILABILITY EACH TYPE BY MONTH:'/
     2         ' ',6X,6(A10,1X)/' ',6X,6(A10,1X))

31300 FORMAT(' ',6X,6(G9.2,1X)/' ',6X,6(G9.2,1X))



      RETURN
      END




      SUBROUTINE ROUTN6(WRT)
      RETURN
      END






      SUBROUTINE PRICE1 (WRT,RED,HREAD,RUN2,RUN3)
      RETURN
      END




      SUBROUTINE PRICE2 (WRT,RED,HREAD,RUN2,RUN3)
      RETURN
      END




      SUBROUTINE PRICE3 (WRT,RED,HREAD,RUN2,RUN3)
      RETURN
      END




      SUBROUTINE PRICE4 (WRT,RED,HREAD,RUN2,RUN3)
      RETURN
      END




      SUBROUTINE PRICE5 (WRT,RED,HREAD,RUN2,RUN3)
C     ---------------------------------

C  ORGNSM
C  ------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE WITH ORGANISM
C  VARIABLES NEED BE RETURNED (ONLY IF PRICE OPTION CHOSEN):
C    NUMBER OF TYPES -    NOOFT(5)
C    NAME OF EACH TYPE - NAMET(5,T)
C    PRICE PER LITRE -    PRICET(5,T,M)
C    AMOUNT AVAILABLE-    N5AVAL(T,M)
C  CURRENT ORGANISM:  BIVALVE

C  STRUCTURE:  MAIN SUBROUTINE, PRICE5, AND 8 LESSER SUBROUTINES
C              MAIN SUBROUTINE DOES THE FOLLOWING:
C               -CALLS SUBROUTINE RIN$5, ROUT$5
C               -LOOPS THROUGH PRODUCTION MONTHS
C                   LOOPS THROUGH PRODUCTION STAGES
C                     CALCULATES AVAILABLE ALGAE QUANTITY
C                     UPDATE2S CURRENT PRODUCTION MONTH MJ
C                     CALLS SUBROUTINE ELEC TO FIND KWHR
C                     CALLS SUBROUTINE FUEL TO FIND LITRES OF OIL
```

```
C                       CALLS SUBROUTINE LABOR TO FIND HOURS OF LABOR
C                       CALLS SUBROUTINE GOODS TO FIND MATERIAL UNITS
C                       CALLS SUBROUTINE COST TO FIND COST OF STAGE
C                       UPDATE2S STAGE TO S+1
C                     CALLS SUBROUTINE WRT$5 -PRINT BY MONTH BATCH BEGUN
C                     UPDATE2S MONTH TO M+1
C                 -ALLOWS PROGRAM TO BE RERUN WITH A FEW DATA CHANGES


C ??????????????????????????????????????????????????????????????????????
C  PRICE OF ALGAE SUBROUTINES - GENERAL INFORMATION
C  ----------------------------------------------

C  NAME      ASSOCIATED FILES USED IN PRICE5
C  ------    ---------------------------------------------------------
C  ICN5$     FOR INCLUSION OF COMMON BLOCKS USED IN PRICE5
C  -------------------------------------------------------------------


C  NAME   DESCRIPTION OF ASSOC. SUBROUTINES, SPECIFIC TO ORGANISM
C  ------ -----------------------------------------------------------
C  RIN$5  READS IN ALL DATA ASSOCIATED WITH PRICE OF INPUT 5
C  ROUT$5 READS OUT ALL DATA READ IN IN RIN $5
C  ELEC   CALCULATES AMOUNT OF ELECTRICITY USED
C  FUEL   CALCULATES AMOUNT OF FUEL OIL USED
C  LABOR  CALCULATES QUANTITY OF LABOR USED
C  GOODS  CALCULATES QUANTITY OF MATERIALS USED
C  COST   COMPUTES COST OF INPUTS USED FOR ALGAE PRODUCTION
C  WRT$5  WRITES OUT PRICE OF ALGAE COMPUTATIONS
C-------------------------------------------------------------------


C  NAME   TYPE  COMMON DIM   DESCRIPTION OF COMMON BLOCK VARIABLE
C  ------ ----  ------ ---   -----------------------------------------
C  ALFOOU R     CN5$  MAXT   AVERAGE LITRES USEABLE BY ANIMALS
C  ALLAB  R     CN5$  MAXT   TOTAL LABOR PER STAGE, LESS INNOCULATION
C  BEGPRO I     CN5$         NUMBER OF MONTH IN WHICH PRODUCTION BEGINS
C  BLBLIF R     CN5$         AVERAGE LIFETIME OF A LIGHTBULB
C  BLBRM  R     CN5$  MAXT   NUMBER OF LIGHTBULBS IN ROOM OF STAGE
C  CALIN  R     CN5$  MAXT   INPUT COST, I, IN STAGE, LESS INNOCULATION
C  CINP   R     CN5$  4      INPUT COST, I, PER UNIT IN STAGE, LESS INOC
C                     MAXT
C  COMEFF R     CN5$         FRACTION EFFICIENCY OF AIR COMPRESSOR
C  CPERML R     CN5$  MAXT   AVERAGE DENSITY OF MATURE CULTURE PER STAGE
C  DATE2  C*25  CN5$         DATE OF ALGAE PRICE DATA
C  DAYSTG R     CN5$  MAXT   DAYS REQUIRED TO COMPLETE A STAGE
C  DISSTG R     CN5$  MAXT   AVERAGE UNAVOIDABLE DISCARD PER STAGE
C  ENDPRO I     CN5$         MONTH IN WHICH PRODUCTION ENDS
C  HPCOM  R     CN5$         AIR COMPRESSOR INPUT POWER RATING
C  HPPMP  R     CN5$         PUMP INPUT POWER RATING
C  HRLABS R     CN5$  6      LABOR DEVOTED TO PRODUCTION OF A STAGE
C                     MAXT
C  HRLABW R     CN5$  6      TYPE L LABOR USE PER WEEK
C  HRSAIR R     CN5$  12     TIME A/C IS ON PER DAY IN STAGE ROOM
C                     MAXT
C  HRSAUT R     CN5$  MAXT   TIME AUTOCLAVE IS ON PER STAGE
C  HRSBLB R     CN5$  MAXT   TIME LIGHTS ARE ON IN STAGE ROOM
C  HRSCOM R     CN5$         TIME COMPRESSOR IS ON PER DAY
C  H2OTNK R     CN5$  MAXT   WATER ADDED AT BEGINNING OF EACH STAGE
C  INQTNK R     CN5$  MAXT   INNOC. VOLUME FOR EACH TANK EACH STAGE
C  LFOOD  R     CN5$  MAXT   ALGAE AVAILABLE FOR FOOD
C  LITWHR R     CN5$  MAXT   ELECTRICAL ENERGY FOR LIGHTING/STAGE
C  MAXLAB I     CN5$         MAXIMUM NUMBER OF LABOR ALLOWED.  PAR=6
C  MONAVA R     CN5$  MAXT   MONTH THAT EACH STAGE IS AVAILABLE TO USE
C  NAMALG C*10  CN5$  MAXT   NAME OF EACH ALGAE STAGE
C  NUMAIR R     CN5$  MAXT   NO. OF AIRCOND. IN ROOM OR EACH STAGE
C  NUMGOD I     CN54         NO. OF DIFFERENT GOOD TYPES, G
C  NUMLAB I     CN5$         NO. OF DIFFERENT LABOR TYPES, I
C  NUMSTG I     CN5$         NO. OF STAGES TO PRODUCE A SINGLE BATCH
C  NUMTNK R     CN5$  MAXT   NUMBER OF TANKS PER STAGE
C  NUMTUB R     CN5$         NUMBER OF TANKS FED BY COMPRESSOR
C  NUTLTR R     CN5$  MAXT   NUTRIENT SOLUTION ADDED PER 1 OF H20/STG
C  PGOOD  R     CN5$  6      PRICE OF GOOD, TYPE G
C  PKWHR  R     CN5$         PRICE PER KWHR OF ELECTRICITY
C  PLABOR R     CN5$  6      PRICE PER HOUR OF TYPE L LABOR
C  PMPEFF R     CN5$         FRACTION EFFICIENCY RATING OF WATER PUMP
C  PMPGPM R     CN5$         OUTPUT CAPACITY RATING OF WATER PUMP
C  POIL   R     CN5$         PRICE PER LITRE OF FUEL OIL
C  QWRT$5 C*1   CN5$         WRITE OUT OUTPUT FOR PRICE5 CALCULATIONS?
C  STGRM  R     CN5$  MAXT   NUMBER OF STAGES IN THE ROOM
C  TALLAB R     CN5$  MAXT   TOTAL TYPE L LABOR PER STAGE, PLUS INNOC.
C  TALLAL R     CN5$  MAXT   TOTAL TYPE L LABOR PER STAGE/LTR, + INNOC
C  TCALIN R     CN5$  MAXT   COST OF ALL INPU, I, PER STAGE, + INNOC.
C  TCELL  R     CN5$  MAXT   TOTAL CELLS OF ALGAE PRODUCED
C  TCINP  R     CN5$  4      COST OF INPUT, I, PER STAGE, PLUS INNOC.
C                     MAXT
C  TCCELL R     CN5$  MAXT   TOTAL COST PER CELL OF ALGAE PRODUCED
C  TCINPL R     CN5$  4      COST OF INPUT, I, PER STAGE, + INNOCULATION
C                     MAXT
C  TCLATY R     CN5$  6      COST OF TYPE L LABOR PER STAGE, + INNOC.
C  TCLITR R     CN5$  MAXT   COST OF ALL INPUT, I, PER LITRE PER STAGE
C  TGOTY  R     CN5$  5      UNITS OF TYPE G GOOD USED PER STAGE
C                     MAXT
C  TKWHR  R     CN5$  MAXT   ELECTRICITY USED PER STAGE, LESS INNOCULA.
C  TLABTY R     CN5$  6      TYPE L LABOR USED PER STAGE, LESS INNOCULA.
C                     MAXT
C  TLITRE R     CN5$  MAXT   ALGAE PRODUCED IN EACH STAGE
C  TOIL   R     CN5$  MAXT   OIL CONSUMPTION PER STAGE, LESS INNOCULATIO
C  TTGOTY R     CN5$  5      TYPE G GOOD USED PER STAGE, PLUS INNOC.
C                     MAXT
C  TTKWHL R     CN5$  MAXT   ELC. PER LITRE PER STAGE, PLUS INNOCULATION
C  TTKWHR R     CN5$  MAXT   ELEC USED PER STAGE, PLUS INNOC.
C  TTLATY R     CN5$  6      LABOR, TYPE LE, PER STAGE, PLUS INNOC.
C                     MAXT
C  TTOIL  R     CN5$  MAXT   OIL CONSUMED PER STAGE, PLUS INNOCULATION
C  TTOILL R     CN5$  MAXT   OIL CONSUMED PER LITRE OF ALGAE
C  TUBTNK R     CN5$  MAXT   COMPRESSOR TUBES PER TANK PER STAGE
C  WATAIR R     CN5$         AIR COMPRESSOR INPUT RATING
C  WATAUT R     CN5$         AUTOCLAVE INPUT WATT RATING
C  WATBLB R     CN5$         LIGHTBULB INPUT RATING
C ---------------------------------------------------------------------
C??????????????????????????????????????????????????????????????????????


C  1. COMMON BLOCK VARIABLES
      INCLUDE (IPAR1)
      INCLUDE (ICN5$)
      INCLUDE (ICUNIT)
      INCLUDE (ICBLK1)
      INCLUDE (ICORG)
      INCLUDE (ICBLKO)
      INCLUDE (ICMAIN)
C  2. ARGUMENT VARIABLES
      INTEGER STAGE, TIME,
     I        WRT, WRT2, RED, RED2, HREAD,
     I        LASSTG
      LOGICAL  RUN2, RUN3
      CHARACTER*9 MONBEG
```

```
C  3. LOCAL VARIABLES
      INTEGER ID, ITDAY, M, MSTART, MSTOP, S
      REAL TDAY, RTDAY
      CHARACTER*1 ANSWR3, QUEST

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ ----------------------------------------------------------
C  ANSWR3 OPTION TO REDO WITH A FEW CHANGES
C  HREAD  ARGUE: IOSTAT TO HELP INDICATE PROBLEMS IN READING FILE
C  ID     ADDITION TO MONTH TO UPDATE FOR CURRENT STAGE
C  ITDAY  INTEGER TO MONTH TO UPDATE FOR CURRENT STAGE
C  LASSTG ARGUE: LAST STAGE OF ALGAE CALCULATED IN THE BATCH
C  M      LOOP COUNTER FOR MONTHS
C  MONBEG ARGUE: MONTH IN WHICH STAGE=1 OF ALGAE IS BEGUN
C  MSTART FIRST MONTH OF ALGAE PRODUCTION TO LOOP THROUGH
C  MSTOP  LAST MONTH OF ALGAE PRODUCTION TO LOOP THROUGH
C  RED    ARGUE: TRANSLATE UNIT READ FROM ORGNSM TO NEXT LEVEL SUBROUTIN
C  RED2   ARGUE: TRANSLATE UNIT READ TO NEXT LEVEL SUBROUTINE
C  RTDAY  CONVERT TOTAL DAYS ELAPSED TO REAL
C  RUN2   ARGUE: INDICATES THAT ONE RUN THRU ALGAE PRICE ALREADY DONE
C  RUN3   ARGUE: INDICATES THAT 1 RUN THRU FROM ABOVE LEVEL SUBR. DONE
C  S      LOOP COUNTER FOR STAGE
C  STAGE  ARGUE: CURRENT STAGE OF ALGAE THAT CALCULATIONS ARE FOR
C  TDAY   TOTAL ELAPSED DAYS
C  TIME   ARGUE: CURRENT MONTH THAT CURRENT STAGE IS IN
C  QUEST  OPTION TO SEE LAST OUTPUT ON SCREEN
C  WRT    ARGUE: TRANSLATE UNIT WRITE FROM ORGANISM TO NEXT LEVEL SUBR.
C  WRT2   ARGUE: UNIT TO WRITE TO NEXT LEVEL SUBROUTINE
C  -----------------------------------------------------------------


C  TRANSLATE ARGUMENTS
      WRT2 = WRT
      RED2 = RED


C  CALL SUBROUTINE TO INPUT ALL DATA
      CALL RIN$5 (WRT2,RED2,RUN2,RUN3)

C  OFFER A SECOND INPUT ENTRY POINT TO ENABLE MINOR INPUT CHANGES
      ANSWR3 = 'X'
 10   IF (ANSWR3 .EQ. 'Y') CALL INPUT2(WRT2,RED2,RUN2,RUN3)

C  OFFER AN OPTION TO PRINT OUTPUT FOR ALGAE PRICE
      WRITE(WI,400)
      READ (RI,1000) QWRT$5
400   FORMAT(' ','DO YOU WANT A PRINTED COPY OF THE ALGAE PRICE ',
     2       'COMPUTATIONS?  Y OR N')


C ASSIGN THE MONTHS TO BE CALCULATED
      MSTART = BEGPRO
      MSTOP = ENDPRO

C  LOOP THROUGH MONTHS OF PRODUCTION - M IS THE MONTH BATCH IS BEGUN
      DO 20 M = MSTART,MSTOP

C        NAME OF THE MONTH IN WHICH PRODUCTION BEGINS
C        ALLOW PRINTING OF AT LEAST ONE STAGE
         MONBEG = NAMEM(M)
         LASSTG = NUMSTG
         MONTH  = M

C        LOOP THROUGH EACH STAGE TO CALCULATE COSTS
         TDAY = 0.
         DO 30 S = 1,NUMSTG

C           ACCUMULATE DAYS TO DETERMINE WHAT MONTH CURRENT STAGE IS
C           BEING PRODUCED IN (FIRST CONVERT TO AN INTEGER)
            STAGE = S
            TDAY = DAYSTG(S) + TDAY
            RTDAY = TDAY + 0.9999999
            ITDAY = RTDAY
            ID = (ITDAY - 1) / 30
            TIME = M + ID
            IF ((ENDPRO-BEGPRO).EQ.11 .AND. TIME.GT.12) TIME = TIME - 12

C           STOP CALCULATIONS IF STAGE CONTINUES INTO NON-PRODUCTION MONTH
            IF (TIME .GT. ENDPRO) THEN
               MONAVA(S) = 'NOT AVAIL.'
               TLITRE(S) = 0.
               LFOOD(S) = 0.
               TCELL(S) = 0.
               IF (S .GT. 1) THEN
                  LFOOD(S-1) = TLITRE(S-1)
                  TCELL(S-1) = CPERML(S-1) * 1000. * TLITRE(S-1)
               ENDIF

C           OTHERWISE CALCULATE ALL STAGE QUANTITIES AND COSTS
            ELSE
C              MONTH THAT ALGAE IS AVAILABLE AND LAST STAGE OF PRODUCTION
               MONAVA(S) = NAMEM(TIME)
               LASSTG = S

C              QUANTITIES PRODUCED AND AVAILABLE -  CHECK FOR LAST STAGE
               TLITRE(S) = (H2OTNK(S)+INQTNK(S)) * NUMTNK(S) - DISSTG(S)
               TCELL(S)  = CPERML(S) * 1000. * TLITRE(S)
               IF (S .EQ. NUMSTG) THEN
                  LFOOD(S) = TLITRE(S)
               ELSE
                  LFOOD(S) = TLITRE(S) - INQTNK(S+1) * NUMTNK(S+1)
               ENDIF

C              CALCULATE INPUT QUANTITIES - BY CALLING SUBROUTINES
               CALL ELEC(STAGE,TIME)
               CALL FUEL (STAGE,TIME)
               CALL LABOR(STAGE,TIME)
               CALL GOODS(STAGE,TIME)

C              CALCULATE INPUT COSTS
               CALL COST(STAGE,TIME)

C              TRANSLATE PRICE5 VARIABLES INTO ORGNSM VARIABLES
              PRICET(5,S,TIME) = TCCELL(S)
              N5AVAL(S,TIME)   = TCELL(S)
            ENDIF

 30      CONTINUE
         IF (QWRT$5 .EQ. 'Y')  CALL WRT$5(MONBEG,LASSTG,WO)

 20   CONTINUE


C  OFFER AN OPTION TO SEE LAST OUTPUT
      WRITE (WI,290)
290   FORMAT (' '/' ','DO YOU WANT TO SEE THE COMPUTATIONS FOR THE  ',
     2       'LAST BATCH OF ALGAE? ')
      READ(RI,1000)  QUEST
```

```
      IF (QUEST .EQ. 'Y')  CALL WRT$5(MONBEG,LASSTG,WI)

C  OFFER AN OPTION TO REDO PROGRAM WITH A FEW INPUT CHANGES
      WRITE (WI,300)
300   FORMAT (' '/' ','DO YOU WANT TO RUN THE ALGAE PRICE COMPUTA',
     2        'TIONS AGAIN'/' ','USING THE SAME DATA WITH A FEW ',
     3        'CHANGES?  Y OR N')
      READ(RI,1000) ANSWR3
      IF (ANSWR3 .EQ. 'Y') GOTO 10
1000  FORMAT (A1)

C  TRANSLATE PRICE5 VARIABLES TO BE RETURNED TO ORGNSM
      NOOFT(5) = NUMSTG
      DO 60 S = 1,NOOFT(5)
         NAMET(5,S) = NAMALG(S)
60    CONTINUE


      RETURN
      END





      SUBROUTINE RIN$5 (WRT,RED,RUN2,RUN3)
C     -------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE WITH ORGANISM
C  VARIABLES NOT RETURNED TO ORGNSM (ONLY TO PRICE5)
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:   READS IN ALL ALGAE PRICE DATA, PRINTS & CORRECTS DATA

C  1. COMMON BLOCK VARIABLES
      INCLUDE (IPAR1)
      INCLUDE (ICN5$)
      INCLUDE (ICUNIT)
      INCLUDE (ICBLK1)
C  2. ARGUMENTS
      INTEGER RED, WRT
      LOGICAL RUN2, RUN3
C  3. LOCAL VARIABLES
      INTEGER G, INFO, K, L, LINNUM, M, NOOFK, S,
     2        R, RETYP1, RETYP2, RETYP3, W
      CHARACTER*1  HALTB
      PARAMETER (NOOFK=38)
      DIMENSION INFO(NOOFK)

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ ---------------------------------------------------------
C  G      LOOP COUNTER FOR GOODS
C  INFO   USED TO ALLOW OR NOT ALLOW DATA ENTRY (0 OR 1)
C  K      LOOP COUNTER FOR DATA LINES
C  L      LOOP COUNTER FOR LABOR
C  LINNUM NUMBER OF THE DATA LINE
C  M      LOOP COUNTER FOR MONTHS
C  NOOFK  NUMBER OF LINES
C  RED    ARGUE:  UNIT NUMBER TO READ
C  R      UNIT NUMBER TO READ
C  RUN2   ARGUE: IS FIRST TIME THROUGH ALGAE PRICE COMPUTATIONS OR NOT?
C  RUN3   ARGUE: INDICATES FROM SUBR. LEVEL ABOVE IF IS 1ST RUN THRU
C  S      LOOP COUNTER FOR STAGES
C  RETYP1 INPUT HEADINGS CONCERNING STAGE
C  RETYP2 INPUT HEADINGS CONCERNING MONTH
C  RETYP3 INPUT HEADINGS CONCERNING
C  WRT    ARGUE: UNIT NUMBER TO WRITE
C  W      UNIT NUMBER TO WRITE
C  -------------------------------------------------------------------------

C  IF NOT FIRST RUN THRU, DO NOT REDO ALL LINES
      IF (RUN2  .OR.  RUN3)  THEN
         W = WI
         R = RI
         DO 120  K=1,NOOFK
               INFO(K) = 1
120      CONTINUE
         RETYP1 = 1
         RETYP2 = 1
         RETYP3 = 1

      ELSE
C     TRANSLATE UNIT NUMBERS TO THOSE OF ORGNSM
         R = RED
         W = WRT
         DO 130 K=1,NOOFK
            INFO(K) = 0
130      CONTINUE
         RETYP1 = 0
         RETYP2 = 0
         RETYP3 = 0
         WRITE(W,2000)
      ENDIF


C  QUE AND READ IN ALL DATA
C  LINE 1-6
 01   IF (INFO( 1) .EQ. 0) THEN
        WRITE(W,201)
        READ(R,*,ERR= 1)DATE2
      ENDIF
 02   IF (INFO( 2) .EQ. 0) THEN
        WRITE(W,202)
        READ(R,*,ERR=2) PKWHR
      ENDIF
 03   IF (INFO( 3) .EQ. 0) THEN
        WRITE(W,203)
        READ(R,*,ERR= 3) POIL
      ENDIF
 04   IF (INFO( 4) .EQ. 0) THEN
        WRITE(W,204)
        READ(R,*,ERR= 4) (PGOOD(G), G=1,NUMGOD)
      ENDIF
 05   IF (INFO( 5) .EQ. 0) THEN
        WRITE(W,205)
        READ(R,*,ERR= 5) NUMLAB
      ENDIF
 06   IF (INFO( 6) .EQ. 0) THEN
        WRITE(W,2206)
        DO 80 L=1,NUMLAB
        WRITE(W,206) L
        READ(R,*,ERR=6) PLABOR(L)
 80     CONTINUE
      ENDIF
```

```
2000  FORMAT(' '/' ','ENTER DATA CONCERNING ALGAE PRODUCTION')
 201  FORMAT(' ','1. DATE OF COMPUTATIONS (TO 25 CHARACTERS IN QUOTES)')
 202  FORMAT(' ','2. PRICE PER KWHR OF ELECTRICITY')
 203  FORMAT(' ','3. PRICE PER LITRE OF FUEL OIL')
 204  FORMAT(' ','4. PRICE PER LIGHTBULB'/
     2       ' ','   PRICE PER LITRE NUTRIENT SOLUTION')
 205  FORMAT(' ','5. NUMBER OF LABOR TYPES - USE AN INTEGER')
2206  FORMAT(' ','6. PRICE PER HOUR OF EACH LABOR TYPE')
 206  FORMAT(' ',2X,'PRICE OF TYPE',I1)

C  LINE NUMBER 7-19
 08   IF (INFO( 8) .EQ. 0) THEN
        WRITE(W,208)
        READ(R,*,ERR= 8) WATAIR
      ENDIF
 09   IF (INFO( 9) .EQ. 0) THEN
        WRITE(W,209)
        READ(R,*,ERR= 9) WATBLB
      ENDIF
 10   IF (INFO(10) .EQ. 0) THEN
        WRITE(W,210)
        READ(R,*,ERR=10) BLBLIF
      ENDIF
 11   IF (INFO(11) .EQ. 0) THEN
        WRITE(W,211)
        READ(R,*,ERR=11) WATAUT
      ENDIF
 12   IF (INFO(12) .EQ. 0) THEN
        WRITE(W,212)
        READ(R,*,ERR=12) HPCOM
      ENDIF
 13   IF (INFO(13) .EQ. 0) THEN
        WRITE(W,213)
        READ(R,*,ERR=13) HRSCOM
      ENDIF
 14   IF (INFO(14) .EQ. 0) THEN
        WRITE(W,214)
        READ(R,*,ERR=14) NUMTUB
      ENDIF
 15   IF (INFO(15) .EQ. 0) THEN
        WRITE(W,215)
        READ(R,*,ERR=15) COMEFF
      ENDIF
 16   IF (INFO(16) .EQ. 0) THEN
        WRITE(W,216)
        READ(R,*,ERR=16) PMPEFF
      ENDIF
 17   IF (INFO(17) .EQ. 0) THEN
        WRITE(W,217)
        READ(R,*,ERR=17) PMPGPM
      ENDIF
 18   IF (INFO(18) .EQ. 0) THEN
        WRITE(W,218)
        READ(R,*,ERR=18) HPPMP
      ENDIF
 19   IF (INFO(19) .EQ. 0) THEN
        WRITE(W,2219)
        DO 81 L=1,NUMLAB
        WRITE(W,219) L
        READ(R,*,ERR=19) HRLABW(L)
 81     CONTINUE
      ENDIF

 208  FORMAT (' ','8.  WATT RATING OF THE AIR CONDITIONER')
```

```
 209  FORMAT (' ','9.  AVERAGE WATT RATING OF LIGHTBULBS')
 210  FORMAT (' ','10. AVERAGE LIFETIME OF LIGHTBULB-IN HOURS')
 211  FORMAT (' ','11. WATT RATING OF AUTOCLAVE OR STERILIZER')
 212  FORMAT (' ','12. HORSEPOWER RATING OF AIR COMPRESSOR')
 213  FORMAT (' ','13. HOURS PER DAY THAT COMPRESSOR OPERATES')
 214  FORMAT (' ','14. TOTAL NUMBER OF AIR COMPRESSOR TUBES')
 215  FORMAT (' ','15. EFFICIENCY RATING OF AIR COMPRESSOR')
 216  FORMAT (' ','16. EFFICIENCY RATING OF WATER PUMP')
 217  FORMAT (' ','17. GALLONS/MINUTE RATING OF WATER PUMP')
 218  FORMAT (' ','18. HORSEPOWER RATING OF WATER PUMP')
2219  FORMAT (' ','19. HOURS PER WEEK OF EACH LABOR TYPE')
 219  FORMAT (' ',4X,'HOURS OF TYPE',I1)

C  LINES 20-35
 20   DO 4000 K=20,35
        IF (INFO(K) .EQ. 0) RETYP1=0
4000  CONTINUE
      IF (INFO(20) .EQ. 0) THEN
        WRITE(W,220)  MAXT
        READ(R,*,ERR=20) NUMSTG
      ENDIF
      IF (RETYP1 .EQ. 0) THEN
        WRITE(W,2222)
        WRITE(W,2223) (S, S=1,NUMSTG)
      ENDIF
 21   IF (INFO(21) .EQ. 0) THEN
        WRITE(W,221)
        READ(R,*,ERR=21) (NUTLTR(S), S=1,NUMSTG)
      ENDIF
 22   IF (INFO(22) .EQ. 0) THEN
        WRITE(W,222)
        READ(R,*,ERR=22) (NUMTNK(S), S=1,NUMSTG)
      ENDIF
 23   IF (INFO(23) .EQ. 0) THEN
        WRITE(W,223)
        READ(R,*,ERR=23) (STGRM (S), S=1,NUMSTG)
      ENDIF
 24   IF (INFO(24) .EQ. 0) THEN
        WRITE(W,224)
        READ(R,*,ERR=24) (DAYSTG(S), S=1,NUMSTG)
      ENDIF
 25   IF (INFO(25) .EQ. 0) THEN
        WRITE(W,225)
        READ(R,*,ERR=25) (CPERML(S), S=1,NUMSTG)
      ENDIF
 26   IF (INFO(26) .EQ. 0) THEN
        WRITE(W,226)
        READ(R,*,ERR=26) (H2OTNK(S), S=1,NUMSTG)
      ENDIF
 27   IF (INFO(27) .EQ. 0) THEN
        WRITE(W,227)
        READ(R,*,ERR=27) (INQTNK(S), S=1,NUMSTG)
      ENDIF
 28   IF (INFO(28) .EQ. 0) THEN
        WRITE(W,228)
        READ(R,*,ERR=28) (DISSTG(S), S=1,NUMSTG)
      ENDIF
 29   IF (INFO(29) .EQ. 0) THEN
        WRITE(W,229)
        READ(R,*,ERR=29) (TUBTNK(S), S=1,NUMSTG)
      ENDIF
 30   IF (INFO(30) .EQ. 0) THEN
        WRITE(W,230)
        READ(R,*,ERR=30) (BLBRM (S), S=1,NUMSTG)
```

```
      ENDIF
 31   IF (INFO(31) .EQ. 0) THEN
        WRITE(W,231)
        READ(R,*,ERR=31) (HRSBLB(S), S=1,NUMSTG)
      ENDIF
 32   IF (INFO(32) .EQ. 0) THEN
        WRITE(W,232)
        READ(R,*,ERR=32) (HRSAUT(S), S=1,NUMSTG)
      ENDIF
 33   IF (INFO(33) .EQ. 0) THEN
        WRITE(W,233)
        READ(R,*,ERR=33) (NUMAIR(S), S=1,NUMSTG)
      ENDIF
 34   IF (INFO(34) .EQ. 0) THEN
        WRITE(W,2234)
        DO 88 L=1,NUMLAB
        WRITE(W,234) L
        READ(R,*,ERR=34) (HRLABS(L,S), S=1,NUMSTG)
88      CONTINUE
      ENDIF
35    IF (INFO(35) .EQ. 0) THEN
        WRITE(W,235)
        READ(R,*,ERR=35) (NAMALG(S), S=1,NUMSTG)
      ENDIF

 220  FORMAT(' '/' 20. NO. OF SEPERABLE STAGES. INTEGER UP TO ',I2)
 2222 FORMAT(' '/' FOR THE FOLLOWING, AN ENTRY IS REQUIRED FOR EACH',
     2       ' ALGAE STAGE.')
 2223 FORMAT(' ',1X,6(I2,'STAGE',2X) / ' ',1X,6(I2,'STAGE',2X))
 221  FORMAT(' ', '21. LITRES NUTRIENT SOL./LITRE H2O')
 222  FORMAT(' ', '22. #OF TANKS')
 223  FORMAT(' ', '23. #OF STAGES IN ROOM')
 224  FORMAT(' ', '24. DAYS TO MATURITY')
 225  FORMAT(' ', '25. CELLS/MILLITRE')
 226  FORMAT(' ', '26. L. H2O ADDED/TANK')
 227  FORMAT(' ', '27. L. INNOCULA/TANK')
 228  FORMAT(' ', '28. L. DISCARD')
 229  FORMAT(' ', '29. # TUBES/TANK')
 230  FORMAT(' ', '30. # LITEBULB/ROOM')
 231  FORMAT(' ', '31. HRS LITES ON/DAY/RM')
 232  FORMAT(' ', '32. HRS AUTOCLAVE')
 233  FORMAT(' ', '33. # AC.S /ROOM')
 2234 FORMAT(' ', '34. HRS LABOR/STAGE')
 234  FORMAT(' ', 4X, 'OF TYPE',I1)
 235  FORMAT(' ','35. NAME OF EACH ALGAE STAGE'/' ',5X,
     2       'UP TO 10 CHARACTERS IN QUOTES')

C LINES 35-41
 36   DO 4001 K=36,41
        IF (INFO(K) .EQ. 0) RETYP2=0
 4001   CONTINUE
      IF (INFO(36) .EQ. 0) THEN
        WRITE(W,236)
        READ(R,*,ERR=36) BEGPRO
      ENDIF
 37   IF (INFO(37) .EQ. 0) THEN
        WRITE(W,237)
        READ(R,*,ERR=37) ENDPRO
      ENDIF
 38   IF (INFO(38) .EQ. 0) WRITE(W,2237) (S, S=1,NUMSTG)
      IF (INFO(38) .EQ. 0) THEN
        WRITE(W,2238)
        DO 82 M=BEGPRO,ENDPRO
        WRITE(W,238) M, NAMEM(M)
        READ(R,*,ERR=38) (HRSAIR(M,S), S=1,NUMSTG)
 82     CONTINUE
      ENDIF

 236  FORMAT (' ','36. THE NUMBER OF THE MONTH YOU BEGIN PRODUCTION'/
     2        ' ','USE AN INTEGER')
 237  FORMAT (' ','37. THE NUMBER OF THE MONTH YOU END PRODUCTION'/
     2        ' ','USE AN INTEGER')
 2237 FORMAT (' ','ENTER INFORMATION CONCERNING EACH STAGE FOR EACH'/
     2        ' ','REQUESTED MONTH'/
     3        ' ',1X,6(I2,'STAGE',2X), / ' ',1X,6(I2,'STAGE',2X))
 2238 FORMAT (' ','38. HRS AC ON/ROOM')
 238  FORMAT (' ',4X,'IN MONTH',I2,2X,A)


C  OPTION TO PRINT OR WRITE OUT DATA INPUT
      CALL ROUT$5

C  OPTION TO DELAY OR TERMINATE
      IF(RETYP3 .EQ. 0) THEN
      WRITE(WI,400)
 400  FORMAT(' '/' ','ENTER S TO STOP EXECUTION  OR,'/
     2          ' ','ENTER C TO CONTINUE EXECUTION'/'  WITH CORRECT',
     3          'IONS LINE BY LINE IN THE ALGAE PRICE DATA')
      READ(RI,1000) HALTB
1000  FORMAT (A1)
      IF (HALTB .EQ. 'S') STOP 'PROGRAM TERMINATED IN ALGAE PRICE'
      ENDIF

C  OPTION TO REINPUT DATA
C  CREATE ARRAY INPUT(I) AND INITIALIZE AT 0.  WHEN DATA IS INPUT,
C  INPUT(I) IS SET = 1.  TO REDO AN INPUT LINE, RESET AT 0.
      DO 86 K=1,NOOFK
        INFO(K) = 1
 86   CONTINUE
      RETYP1 = 1
      RETYP2 = 1
      RETYP3 = 1

C  RETURN FROM PRICE5 SUBROUTINE TO CHANGE PRICE DATA
      ENTRY INPUT2(WRT,RED,RUN2,RUN3)
      W = WI
      R = RI

      WRITE(WI,500) NOOFK
 500  FORMAT(' '/' ','YOU MAY CORRECT DATA A LINE AT A TIME FOR ',
     2       'ALGAE PRICE'/' ',2X,'ENTER  0  TO MAKE NO CORRECTIONS ',
     3       'OR'/' ',2X,'ENTER THE NUMBER OF THE LINE YOU WISH TO ',
     4       'CORRECT'/' ','ENTER AN INTEGER 0 TO ',I2)
 87   CONTINUE
      READ(RI,*) LINNUM
      IF (LINNUM.LT.0 .OR. LINNUM.GT.NOOFK ) THEN
        WRITE(WI,*) 'WRONG ENTRY - RETYPE'
        GOTO 87
      ELSE IF (LINNUM .NE. 0) THEN
        INFO(LINNUM) = 0
        GOTO (1,2,3,4,5,6,6,8,9,10,11,12,13,14,15,16,17,18,19,
     B        20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,
     C        36,36,36),LINNUM
      ENDIF


      RETURN
```

```
      END




      SUBROUTINE ROUT$5
C     -----------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE WITH ORGANISM
C  VARIABLES NOT RETURNED TO ORGNSM, ONLY TO PRICE5
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:   READS OUT DATA RIN$5 IN SUBROUTINE INPUT,
C               WHICH IS USED TO CALCULATE PRICE5, ALGAE

C 1. COMMON BLOCKS
      INCLUDE (IPAR1)
      INCLUDE (ICN5$)
      INCLUDE (ICUNIT)
      INCLUDE (ICBLK1)
      INCLUDE (ICBLKO)
      INCLUDE (ICMAIN)
C 2. LOCAL VARIABLES
      CHARACTER*1 COPYA, COPYB
      INTEGER PRINT1, PRINT2, G, L, P, M, S, W

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ -------------------------------------------------------------
C  COPYA  OPTION TO PRINT INPUT DATA
C  COPYB  OPTION TO WRITE INPUT DATA
C  G      LOOP COUNTER FOR GOODS
C  L      LOOP COUNTER FOR GOODS
C  M      LOOP COUNTER FOR MONTHS
C  P      LOOP COUNTER FOR DIFFERENT PRINTING OPTIONS
C  PRINT1 PRINTING INITIAL VALUE (1 OR 2)
C  PRINT2 PRINTING FINAL VALUE (1 OR 2)
C  S      LOOP COUNTER STAGE
C  W      UNIT NUMBER TO WRITE
C  -------------------------------------------------------------------


C TRANSLATE UNIT NUMBERS TO UNIT NUMBERS USED IN ORGNSM
      W = WO

C OPTION TO PRINT INPUT
      WRITE(WI,3000)
3000  FORMAT(' '/' ','DO YOU WANT TO PRINT THE ALGAE PRICE DATA ',
     2       'ENTERED?')
      READ(RI,1000) COPYA
1000  FORMAT (A1)
      WRITE(WI,3001)
3001  FORMAT(' ','DO YOU WANT TO SEE THE ALGAE DATA ON THE SCREEN? ',
     2       'Y OR N')
      READ(RI,1000) COPYB

      IF (COPYA .EQ. 'Y' .AND.  COPYB .NE. 'Y') THEN
        PRINT1 = 1
        PRINT2 = 1
        ENDIF
      IF (COPYB .EQ. 'Y'  .AND.  COPYA .NE. 'Y') THEN
        PRINT2 = 2
        ENDIF
      IF (COPYA .EQ. 'Y'  .AND.  COPYB .EQ. 'Y') THEN
        PRINT1 = 1
        PRINT2 = 2
        ENDIF
      IF (COPYA .NE. 'Y'  .AND.  COPYB .NE. 'Y') GOTO 98


      DO 99 P = PRINT1,PRINT2
        IF (P .EQ. 2) V = WI
        WRITE(W,3300)  NAME, DATE
        WRITE(W,301) DATE2
        WRITE(W,3302)
        WRITE(W,302) PKWHR
        WRITE(W,303) POIL
        WRITE(W,304) (PGOOD(G), G=1,NUMGOD)
        WRITE(W,305) NUMLAB
        DO 83 L=1,NUMLAB
          WRITE(W,306) L, PLABOR(L)
 83       CONTINUE

        WRITE(W,3307)
        WRITE(W,308) WATAIR
        WRITE(W,309) WATBLB
        WRITE(W,310) BLBLIF
        WRITE(W,311) WATAUT
        WRITE(W,312) HPCOM
        WRITE(W,313) HRSCOM
        WRITE(W,314) NUMTUB
        WRITE(W,315) COMEFF
        WRITE(W,316) PMPEFF
        WRITE(W,317) PMPGPM
        WRITE(W,318) HPPMP
        WRITE(W,3319)
        DO 84 L=1,NUMLAB
           WRITE(W,319) L, HRLABW(L)
 84     CONTINUE

        WRITE(W,3320)
        WRITE(W,320)  NUMSTG
        WRITE(W,3321)(S        , S=1,NUMSTG)
        WRITE(W,321) (NUTLTR(S), S=1,NUMSTG)
        WRITE(W,322) (NUMTNK(S), S=1,NUMSTG)
        WRITE(W,323) (STGRM (S), S=1,NUMSTG)
        WRITE(W,324) (DAYSTG(S), S=1,NUMSTG)
        WRITE(W,325) (CPERML(S), S=1,NUMSTG)
        WRITE(W,326) (H2OTNK(S), S=1,NUMSTG)
        WRITE(W,327) (INQTNK(S), S=1,NUMSTG)
        WRITE(W,328) (DISSTG(S), S=1,NUMSTG)
        WRITE(W,329) (TUBTNK(S), S=1,NUMSTG)
        WRITE(W,330) (BLBRM (S), S=1,NUMSTG)
        WRITE(W,331) (HRSBLB(S), S=1,NUMSTG)
        WRITE(W,332) (HRSAUT(S), S=1,NUMSTG)
        WRITE(W,333) (NUMAIR(S), S=1,NUMSTG)
        WRITE(W,3334)
        DO 89 L=1,NUMLAB
          WRITE(W,334)L,(HRLABS(L,S), S=1,NUMSTG)
 89     CONTINUE
        WRITE(W,335) (NAMALG(S), S=1,NUMSTG)

        WRITE(W,3336)
        WRITE(W,336) BEGPRO, NAMEM(BEGPRO)
        WRITE(W,337) ENDPRO, NAMEM(ENDPRO)
```

```
      WRITE(W,3338) (S, S=1,NUMSTG)
      WRITE(W,3339)
      DO 85 M=BEGPRO,ENDPRO
        WRITE(W,338) M, NAMEM(M), (HRSAIR(M,S), S=1,NUMSTG)
85      CONTINUE

C     RESET UNIT NUMBER
      W = WI

99    CONTINUE


3300 FORMAT('1',A,' COST SIMULATION MODEL:  ALGAE PRICE COMPUTATIONS'/
   2        ' ',59('-')/' DATE:  ',A25/' INPUTS ENTERED')
301  FORMAT(' ','1.',2X,A,' (DATE OF PRICE COMPUTATIONS)'/)
3302 FORMAT('0','*PRICE INFORMATION')
302  FORMAT(' ','2. OF ELECTRICITY'  ,T40,'$',T42,F8.4,T53,'PER KWHR')
303  FORMAT('0','3. OF FUEL OIL'     ,T40,'$',T42,F8.4,T53,'PER LITR')
304  FORMAT('0','4. OF LIGHTBULBS   ',T40,'$',T42,F8.4,T53,'PER BULB'/
   B        '0','4. OF NUTRIENT SOL.',T40,'$',T42,F8.4,T53,'PER LITR')
305  FORMAT('0','5. # OF LABOR TYPES:',1X,I2)
306  FORMAT(' ','6. PRICE OF TYPE',I1,T40,'$',T42,F8.4,T53,'PER HR')

3307 FORMAT('0','*MISC. INFORMATION - NON STAGE DEPENDENT')
308  FORMAT('0','8.  WATT RATING OF AIR CONDIT.',T40,F10.1 ,T56,'WATT')
309  FORMAT('0','9.  WATT RATING OF LITEBULBS'  ,T40,F10.1 ,T56,'WATT')
310  FORMAT('0','10. LIFETIME OF LITEBULBS'     ,T40,F10.1 ,T56,'HRS ')
311  FORMAT('0','11. WATT RATING OF AUTOCLAVE'  ,T40,F10.1 ,T56,'WATT')
312  FORMAT('0','12. COMPRESSOR HORSEPOWER'     ,T40,F10.1 ,T56,'HP  ')
313  FORMAT('0','13. HOURS COMPRESSOR ON/DAY'   ,T40,F10.1 ,T56,'HRS ')
314  FORMAT('0','14. TOTAL # COMPRESSOR TUBES'  ,T40,F10.0 ,T56,'TUBE')
315  FORMAT('0','15. EFF. RATING OF COMPRESSOR' ,T40,F10.3 ,T56)
316  FORMAT('0','16. EFF. RATING OF WATER PUMP' ,T40,F10.3 ,T56)
317  FORMAT('0','17. GPM RATING OF WATER PUMP'  ,T40,F10.1 ,T56,'GPM ')
318  FORMAT('0','18. HP RATING OF WATER PUMP'   ,T40,F10.1 ,T56,'HP  ')
3319 FORMAT('0','19. HOURS OF LABOR/WEEK')
319  FORMAT(' ',4X,'OF TYPE',I1                 ,T40,F10.1 ,T56,'HRS ')

3320 FORMAT('0','*STAGE INFORMATION')
320  FORMAT('0','20. NUMBER OF STAGES:',1X,I2)
3321 FORMAT('0',T43,6(3X,I2,'STAGE',5X)/' ',T43,6(I2,'STAGE',5X)//)
321  FORMAT('0','21. NUTRIENT/LITRE H2O',
   B        T41, 6(G13.3,2X)/ ' ', T41, 6(G13.3,2X))
322  FORMAT('0','22. NUMBER OF TANKS',
   B        T41, 6(F13.1,2X)/ ' ', T41, 6(F13.1,2X))
323  FORMAT('0','23. NUMBER OF STAGES PER ROOM',
   B        T41, 6(F13.1,2X)/ ' ', T41, 6(F13.1,2X))
324  FORMAT('0','24. DAYS TO REACH MATURITY',
   B        T41, 6(F13.1,2X)/ ' ', T41, 6(F13.1,2X))
325  FORMAT('0','25. CELLS PER MILLITRE - DENSITY',
   B        T41, 6(G13.3,2X)/ ' ', T41, 6(G13.3,2X))
326  FORMAT('0','26. LITRES WATER ADDED PER TANK',
   B        T41, 6(G13.3,2X)/ ' ', T41, 6(G13.3,2X))
327  FORMAT('0','27. LITRES OF INNOCULATION ADDED/TANK',
   B        T41, 6(G13.3,2X)/ ' ', T41, 6(G13.3,2X))
328  FORMAT('0','28. AVG. LITRES UNCONTROLLED DISCARD',
   B        T41, 6(G13.3,2X)/ ' ', T41, 6(G13.3,2X))
329  FORMAT('0','29. NUMBER OF AIR TUBES PER TANK',
   B        T41, 6(G13.3,2X)/ ' ', T41, 6(G13.3,2X))
330  FORMAT('0','30. NUMBER OF LIGHTBULBS PER ROOM',
   B        T41, 6(F13.1,2X)/ ' ', T41, 6(F13.1,2X))
331  FORMAT('0','31. HOURS LIGHTS ON PER DAY PER ROOM',
   B        T41, 6(F13.2,2X)/ ' ', T41, 6(F13.2,2X))
332  FORMAT('0','32. HOURS AUTOCLAVE USED',
   B        T41, 6(F13.2,2X)/ ' ', T41, 6(F13.2,2X))
333  FORMAT('0','33. NUMBER OF AIR CONDIT. PER ROOM',
   B        T41, 6(F13.1,2X)/ ' ', T41, 6(F13.1,2X))
3334 FORMAT('0','34. HRS OF LABOR PER STAGE')
334  FORMAT(' ',4X,'OF TYPE',I1,T41, 6(F13.2,2X)/' ',T41, 6(F13.2,2X))
335  FORMAT('0','35. NAME OF EACH ALGAE STAGE',
   2         T41,6(A10,5X)/' ',T41,6(A10,5X))

3336 FORMAT ('0','*MONTH INFORMATION')
336  FORMAT ('0','36. MONTH BEGIN PRODUCTION',2X,I2,1X,A)
337  FORMAT ('0','37. MONTH END PRODUCTION',4X,I2,1X,A)
3338 FORMAT ('0',T43 ,6(3X,I2,'STAGE',5X)/ ' ',T43,6(3X,I2,'STAGE',5X))
3339 FORMAT ('0','38. HOURS AIR CONDIT. ON PER DAY/ROOM')
338  FORMAT ('0',3X,I2,1X,A,T41,6(F13.2,2X)/ ' ',T41,6(F13.2,2X))

98   CONTINUE


     RETURN
     END






     SUBROUTINE ELEC(STAGE,TIME)
C    ---------------------------

C ORGNSM:
C -------
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE, RETURNED ONLY TO PRICE5
C CURRENT ORGANISM:  BIVALVES

C STRUCTURE:  CALCULATES WHR FOR MAJOR ELECTRICAL USE, AND SUMS FOR
C             TOTAL KWHR PER STAGE, PER LITRE OF ALGAE PRICE PRODUCTION

C 1. COMMON BLOCK VARIABLES
     INCLUDE (IPAR1)
     INCLUDE (ICN5$)
     INCLUDE (ICUNIT)
C 2. ARGUMENT VARIABLES
     INTEGER STAGE, TIME
C 3. LOCAL VARIABLES
     INTEGER MJ, S
     REAL AIRWHR, AUTWHR, COMIN, COMWHR, INQKWH, PERCOM,
   2      PMPCON, PMPIN, PMPINL, PMPWHR, WATHP

C NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C ------ -----------------------------------------------------------
C AIRWHR ELEC. ENERGY FOR AIR CONDITIONING
C AUTWHR ELEC. ENERGY FOR STERILIZING
C COMIN  ELEC INPUT FOR AIR COMPRESSOR
C COMWHR ELEC ENERGY FOR AIR COMPRESSOR
C INQKWH ELEC ENERGY USED FOR INNOCULATION
C MJ     MONTH IN WHICH CURRENT STAGE IS PRODUCED
C PERCOM FRACTION OF TOTAL AIR COMPRESSOR USED/DAY
C PMPCON CONVERSION FACTOR FOR GAL/MIN TO LITRE/MIN
C PMPIN  PUMP ELECTRICAL INPUT
C PMPINL ELEC POWER INPUT TO PUMP/LITRE
C PMPWHR ELEC. ENERGY TO PUMP ALL WATER/STAGE
C S      CURRENT STAGE
C STAGE  ARGUE:  TO TRANSLATE CURRENT STAGE OF ALGAE, S
```

```
C  TIME   ARGUE:  TO TRANSLATE CURRENT MONTH STAGE IS IN TO MJ
C  WATHP  CONVERSION FACTOR FOR WATTS/HP
C  -----------------------------------------------------------------

C  INITIALIZE LOCAL
      DATA WATHP/ 745.7 /
C     CONVERT PUMP FROM GPM TO L/HR 3.78537L/G * 60M/H
      DATA PMPCON/ 227.122/

C  TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
      S = STAGE
      MJ = TIME


C  LIGHTING
      LITWHR(S) = (WATBLB * BLBRM(S) * HRSBLB(S) / STGRM(S)) * DAYSTG(S)

C  AUTOCLAVE
      AUTWHR = WATAUT * HRSAUT(S)

C  AIRCOMPRESSOR: TST FOR 0, FIND MACHINE INPUT, FIND % TOTAL MACHINE
      IF (COMEFF.EQ.0. .OR. NUMTUB.EQ.0.) THEN
        COMWHR = 0.
      ELSE
        COMIN = (HPCOM * WATHP * HRSCOM) / COMEFF
        PERCOM = (TUBTNK(S) * NUMTNK(S)) / NUMTUB
        COMWHR = COMIN * PERCOM * DAYSTG(S)
      ENDIF

C  PUMPING: TEST FOR 0, FIND MACHINE INPUT, FIND MACHINE INPUT PER
C           LITRE, AND FIND INPUT FOR ALL REQUIRED WATER
      IF (PMPGPM.EQ.0. .OR. PMPEFF.EQ.0.) THEN
        PMPWHR = 0.
      ELSE
        PMPIN = (HPPMP * WATHP) / PMPEFF
        PMPINL = PMPIN / (PMPGPM * PMPCON)
        PMPWHR = PMPINL * H2OTNK(S) * NUMTNK(S)
      ENDIF

C  AIR CONDITIONING
      AIRWHR = (WATAIR * NUMAIR(S) * HRSAIR(MJ,S) /STGRM(S)) *DAYSTG(S)

C  SUM ALL WHR , CHANGE TO KWHR
      TKWHR(S) = (LITWHR(S) + AUTWHR + COMWHR + PMPWHR + AIRWHR) * .001

C  FIND KWHR PLUS INNOCULATION - CHECK FOR FIRST STAGE
      IF (S .EQ. 1) THEN
        INQKWH = 0.
        TTKWHR(S) = TKWHR(S)
      ELSE
        INQKWH = INQTNK(S) * NUMTNK(S) * TTKWHL(S-1)
        TTKWHR(S) = INQKWH + TKWHR(S)
      ENDIF

C  FIND KWHR PER LITRE
      IF (TLITRE(S) .LE. 0.) STOP 'ERROR IN ELEC; TOTAL LITRES = 0'
      TTKWHL(S) = TTKWHR(S) / TLITRE(S)


      RETURN
      END



      SUBROUTINE FUEL(STAGE,TIME)
C     ---------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE, BUT RETURNED ONLY TO PRICE5
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  COMPUTES LITRES OF OIL USED PER STAGE OF ALGAE PRODUCTION

C  1. COMMON BLOCK VARIABLES
      INCLUDE (IPAR1)
      INCLUDE (ICN5$)
C  2. ARGUMENT VARIABLES
      INTEGER STAGE
C  3. LOCAL VARIABLES
      INTEGER S

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ -------------------------------------------------------------
C  S      TRANSLATES ARGUMENT FOR CURRENT STAGE
C  STAGE  CURRENT STAGE OF ALGAE BEING PRODUCED
C  --------------------------------------------------------------------


C  TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
      S = STAGE



C  *** NOTE:  BECAUSE THIS IS ONLY A DUMMY ROUTINE, ACCURATE ALGAE
C  COSTS CANNOT BE COMPUTED FOR MONTHS IN WHICH HEATING
C  WOULD PLAY A SIGNIFICANT ROLE
C  ***

C  FUEL CANNOT BE COMPUTED AT THIS TIME.  SET VARIABLE = 0.
      TOIL(S)  = 0.
      TTOIL(S) = 0.
      RETURN
      END





      SUBROUTINE LABOR(STAGE,TIME)
C     ----------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE, RETURNED ONLY TO PRICE5
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:   CALCULATES THE QUANTITY OF LABOR TYPE L USED
C               PER STAGE, PER LITRE OF ALGAE PRODUCTION

C  1. COMMON BLOCK VARIABLES
      INCLUDE (IPAR1)
      INCLUDE (ICN5$)
C  2. ARGUMENT VARIABLES
```

```
      INTEGER STAGE, TIME
C  3. LOCAL VARIABLES
      INTEGER L, MJ, S
      REAL IQLATY(6)

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ -------------------------------------------------------------
C  IQLATY LABOR TYPE L REQUIRED FOR INNOCULATION
C  L      LOOP COUNTER FOR LABOR TYPES
C  MJ     TRANSLATE ARGUMENT TIME
C  S      TRANSLATES ARGUMENT STAGE FOR USE IN SUBROUTINE
C  STAGE  ARGUE:  CURRENT ALGAE STAGE
C  TIME   ARGUE:  MONTH THAT CURRENT STAGE IS IN
C  --------------------------------------------------------------------

C  TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
      S = STAGE
      MJ = TIME


C  FIND QUANTITY OF LABOR TYPE L, FILE INTO ARRAY
      ALLAB(S) = 0.
      TALLAB(S) = 0.

      DO 10 L = 1,NUMLAB
        TLABTY(L,S) = HRLABW(L)/7. *DAYSTG(S)/FLOAT(NUMSTG) +HRLABS(L,S)

C       ADD INNOCULATION - TEST FOR FIRST STAGE
        IF (S .EQ. 1) THEN
          IQLATY(L) = 0.
          TTLATY(L,S) = TLABTY(L,S)
        ELSE
          IF (TLITRE(S-1) .LE. 0.) STOP 'ERROR LABOR; TLITRE_0'
          IQLATY(L) = INQTNK(S)*NUMTNK(S) * (TTLATY(L,S-1)/TLITRE(S-1))
          TTLATY(L,S) = TLABTY(L,S) + IQLATY(L)
        ENDIF

C       SUM ALL TYPES L
        ALLAB(S) = TLABTY(L,S) + ALLAB(S)
        TALLAB(S) = TTLATY(L,S) + TALLAB(S)

10    CONTINUE

C  TOTAL ALL LABOR USED PER LITRE
      IF (TLITRE(S) .LE. 0.) STOP 'ERROR LABOR2: TLITRE_0'
      TALLAL(S) = TALLAB(S) / TLITRE(S)


      RETURN
      END





      SUBROUTINE GOODS(STAGE,TIME)
C     ----------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLE CHANGE, RETURNED ONLY TO PRICE5
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  COMPUTES QUANTITY OF GOODS USED PER STAGE OF ALGAE PRODUC

C  1. COMMON BLOCK VARIABLES
      INCLUDE (IPAR1)
      INCLUDE (ICN5$)
C  2. ARGUMENT VARIABLES
      INTEGER STAGE, TIME
C  3. LOCAL VARIABLES
      INTEGER G, S, MJ
      REAL IQGOTY(5)

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ -------------------------------------------------------------
C  IQGOTY GOOD TYPE G REQUIRED FOR INNOCULATION
C  G      LOOP COUNTER FOR GOODS
C  MJ     TRANSLATES ARGUMENT TIME FOR USE IN SUBROUTINE
C  S      TRANSLATES ARGUMENT STAGE FOR USE IN SUBROUINTE
C  STAGE  ARGUE:  CURRENT STAGE OF ALGAE
C  TIME   ARGUE:  MONTH THAT CURRENT STAGE IS IN
C  --------------------------------------------------------------------


C  TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
      S = STAGE
      MJ = TIME

C  LIGHTBULBS
      IF (WATBLB.EQ.0. .OR. BLBLIF.EQ.0.) THEN
        TGOTY(1,S) = 0.
      ELSE
        TGOTY(1,S) = LITWHR(S) / WATBLB / BLBLIF
      ENDIF

C  NUTRIENT SOLUTION
      TGOTY(2,S) = NUTLTR(S) * H2OTNK(S) * NUMTNK(S)

C  FIND TOTAL GOOD TYPE G INCLUDING INNOCULATION - CHECK FOR FIRST STAGE
      DO 10 G = 1,NUMGOD
        IF (S .EQ. 1) THEN
          IQGOTY(G) = 0.
          TTGOTY(G,S) = TGOTY(G,S)
        ELSE
          IQGOTY(G) = (INQTNK(S)*NUMTNK(S))*(TTGOTY(G,S-1)/TLITRE(S-1))
          TTGOTY(G,S) = TGOTY(G,S) + IQGOTY(G)
        ENDIF
 10   CONTINUE


      RETURN
      END





      SUBROUTINE COST (STAGE,TIME)
C     ----------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES VARY, BUT RETURNED ONLY TO PRICE5
C  CURRENT ORGANISM:  BIVALVES
```

```
C  SUBROUTINE: DETERMINES COST OF GIVEN ALGAE INPUT QUANT. FOR PRICE N=5

C  1. COMMON BLOCK VARIABLES
      INCLUDE (IPAR1)
      INCLUDE (ICN5$)
C  2. ARGUMENT VARIABLES
      INTEGER TIME, STAGE
C  3. LOCAL VARIABLES
      INTEGER G, I, L, S
      REAL CLABTY, TCGOTY, CGODTY

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ -------------------------------------------------------------
C  CGODTY COST OF GOOD, TYPE G, LESS INNOCULATION
C  CLABTY COST OF LABOR, TYPE L, LESS INNOCULATION
C  G      LOOP COUNTER FOR GOODS
C  I      LOOP COUNTER FOR INPUTS
C  L      LOOP COUNTER FOR LABOR TYPES
C  S      LOOP COUNTER FOR STAGES
C  STAGE  ARGUE:  CURRENT STAGE OF ALGAE
C  TCGOTY TOTAL COST OF GOODS, TYPE G, LESS INNOCULATION
C  TIME   ARGUE: MONTH THAT CURRENT STAGE IS IN
C  --------------------------------------------------------------------


C  TRANSLATE ARGUMENT INTO LOCAL VARIABLES
      S = STAGE
      M = TIME

C  FIND COST OF EACH INPUT TYPE LESS INNOCULATION, FILE INTO ARRAY
      CINP(1,S) = TKWHR(S) * PKWHR
      CINP(2,S) = TOIL(S) * POIL
      CINP(3,S) = 0.
        DO 10 L = 1,NUMLAB
        CLABTY = TLABTY(L,S) * PLABOR(L)
        CINP(3,S) = CLABTY + CINP(3,S)
 10     CONTINUE
      CINP(4,S) = 0.
        DO 20 G = 1,NUMGOD
        CGODTY = TGOTY(G,S) * PGOOD(G)
        CINP(4,S) = CGODTY + CINP(4,S)
 20     CONTINUE

C  FIND THE COST OF EACH INPUT TYPE PLUS INNOCULATION
      TCINP(1,S) = TTKWHR(S) * PKWHR
      TCINP(2,S) = TTOIL(S) * POIL
      TCINP(3,S) = 0
        DO 30 L = 1,NUMLAB
        TCLATY(L) = TTLATY(L,S) * PLABOR(L)
        TCINP(3,S) = TCLATY(L) + TCINP(3,S)
 30     CONTINUE
      TCINP(4,S) = 0.
        DO 40 G = 1,NUMGOD
        TCGOTY = TTGOTY(G,S) * PGOOD(G)
        TCINP(4,S) = TCGOTY + TCINP(4,S)
 40     CONTINUE

C  TOTAL COSTS
      CALIN(S) = 0.
      TCALIN(S) = 0.
      DO 50 I = 1,4

C       COST PER LITRE OF EACH INPUT
        TCINPL(I,S) = TCINP(I,S) / TLITRE(S)

C       SUM OF ALL INPUT COSTS
        CALIN(S) = CINP(I,S) + CALIN(S)

C       SUM OF ALL INPUT COSTS PLUS INNOCULATION
        TCALIN(S) = TCINP(I,S) + TCALIN(S)

 50   CONTINUE

C TOTAL COST PLUS INNOCULATION -  PER LITRE, CELL FOOD
      TCLITR(S) = TCALIN(S) / TLITRE(S)
C     CHECK FOR 0
      IF (TCELL(S) .EQ. 0.) THEN
        TCCELL(S) = 0.
      ELSE
        TCCELL(S)  = TCALIN(S)  / TCELL(S)
      ENDIF


      RETURN
      END






      SUBROUTINE WRT$5(MONBEG,LASSTG,WRT)
C     -----------------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE, RETURNED ONLY TO PRICE5
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  OUTPUTS DATA BY THE MONTH IN WHICH ALGAE STAGE 1 IS BEGUN

C  1. COMMON BLOCK VARIABLES
      INCLUDE (IPAR1)
      INCLUDE (ICBLK1)
      INCLUDE (ICN5$)
      INCLUDE (ICUNIT)
      INCLUDE (ICBLKO)
      INCLUDE (ICMAIN)
C  2. ARGUMENT VARIABLES
      INTEGER LASSTG, WRT
      CHARACTER*9 MONBEG
C  3. LOCAL VARIABLES
      INTEGER S, W

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ -------------------------------------------------------------
C  LASSTG ARGUE: LAST STAGE OF ALGAE THAT CALCULATIONS ARE DONE FOR
C  MONBEG ARGUE: MONTH IN WHICH CURRENT BATCH OF ALGAE WAS BEGUN
C  S      LOOP COUNTER FOR STAGES
C  WRT    ARGUE:  UNIT NUMBER TO WRITE
C  W      UNIT NUMBER TO WRITE
C  --------------------------------------------------------------------


C  TRANSLATE UNIT NUMBERS TO NUMBERS USED IN ORGNSM
      W = WRT
```

```
      WRITE(W,1101) NAME, DATE
      WRITE(W,101) DATE2
      WRITE(W,102) MONBEG
      WRITE(W,1103) (S,           S = 1,NUMSTG)

      WRITE(W,103 ) (MONAVA(  S), S = 1,NUMSTG)
      WRITE(W,1031) (TLITRE(  S), S = 1,LASSTG)
      WRITE(W,104 ) (LFOOD (  S), S = 1,LASSTG)
      WRITE(W,105 ) (TCELL (  S), S = 1,LASSTG)

      WRITE(W,108 ) (TCALIN(  S), S = 1,LASSTG)
      WRITE(W,109 ) (TCLITR(S  ), S = 1,LASSTG)
      WRITE(W,1092) (TCCELL(  S), S = 1,LASSTG)
      WRITE(W,110 ) (CALIN (  S), S = 1,LASSTG)

      WRITE(W,113 ) (TCINP (1,S), S = 1,LASSTG)
      WRITE(W,114 ) (TCINPL(1,S), S = 1,LASSTG)
      WRITE(W,115 ) (CINP  (1,S), S = 1,LASSTG)
      WRITE(W,116 ) (TTKWHR(  S), S = 1,LASSTG)
      WRITE(W,117 ) (TTKWHL(  S), S = 1,LASSTG)
      WRITE(W,118 ) (TKWHR (  S), S = 1,LASSTG)
      WRITE(W,119 ) (TCINP (2,S), S = 1,LASSTG)
      WRITE(W,120 ) (TCINPL(2,S), S = 1,LASSTG)
      WRITE(W,121 ) (CINP  (2,S), S = 1,LASSTG)
      WRITE(W,122 ) (TTOIL (  S), S = 1,LASSTG)
      WRITE(W,123 ) (TTOILL(  S), S = 1,LASSTG)
      WRITE(W,124 ) (TOIL  (  S), S = 1,LASSTG)

      WRITE(W,125 ) (TCINP (3,S), S = 1,LASSTG)
      WRITE(W,126 ) (TCINPL(3,S), S = 1,LASSTG)
      WRITE(W,127 ) (CINP  (3,S), S = 1,LASSTG)
      WRITE(W,128 ) (TALLAB(  S), S = 1,LASSTG)
      WRITE(W,129 ) (TALLAL(  S), S = 1,LASSTG)
      WRITE(W,130 ) (ALLAB (  S), S = 1,LASSTG)
      WRITE(W,131 ) (TCINP (4,S), S = 1,LASSTG)
      WRITE(W,132 ) (TCINPL(4,S), S = 1,LASSTG)
      WRITE(W,133 ) (CINP  (4,S), S = 1,LASSTG)


1101 FORMAT ('1'//' ',A10,' COST SIMULATION MODEL:  ALGAE PRICE ',
    2        'COMPUTATIONS'/' ',59('-')/' DATE:  ',A25/
    3        ' OUTPUT GENERATED')
101  FORMAT('0','DATE OF PRICE COMPUTATIONS:  ',A///)
102  FORMAT ('0', 'BATCH BEGUN:',2X, A/)
1103 FORMAT('0',T43,6(3X,I2,'STAGE',5X)/' ',T43,6(3X,I2,'STAGE',5X)//)

103  FORMAT ('0','ALGAE AVAILABLE FOR USE IN MONTH',
    B           T41, 6(4X,A9,2X)/ ' ', T41, 6(4X,A9,2X))
1031 FORMAT ('0', '*TOTAL LITRES PRODUCED',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
104  FORMAT ('0', 'LITRES OF ALGAE AVAILABLE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
105  FORMAT ('0', 'CELLS OF ALGAE PRODUCED',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
108  FORMAT ('0', '*TOTAL COST OF STAGE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
109  FORMAT ('0', 1X, 'TOTAL COST PER LITRE OF ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
1092 FORMAT ('0',1X,'TOTAL COST PER CELL OF ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
110  FORMAT ('0', 1X, 'TOTAL COST OF STAGE, LESS INNOCULATION',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))

113  FORMAT ('0', '*TOTAL COST OF ELECTRICITY',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
114  FORMAT ('0', 1X, 'TOTAL COST OF ELECTRICITY/LITRE OF ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
115  FORMAT ('0', 1X, 'TOTAL COST OF ELEC. LESS INNNOCULATION',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
116  FORMAT ('0', '*TOTAL KWHR USED',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
117  FORMAT ('0', 1X, 'TOTAL KWHR USED PER LITRE OF ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
118  FORMAT ('0', 1X, 'TOTAL KWHR USED LESS INNOCULATION',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))

119  FORMAT ('0', '*TOTAL COST OF FUEL OIL',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
120  FORMAT ('0', 1X, 'TOTAL COST OF OIL/LITRE OF ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
121  FORMAT ('0', 1X, 'TOTAL COST OF OIL LESS INNOCULATION',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
122  FORMAT ('0', '*TOTAL LITRES OF FUEL OIL',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
123  FORMAT ('0', 1X, 'TOTAL LITRES OIL USED/LITRE OF ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
124  FORMAT ('0', 1X, 'TOTAL L. OIL USED LESS INNOCULATION',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))

125  FORMAT ('0', '*TOTAL COST OF ALL LABOR TYPES',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
126  FORMAT ('0', 1X, 'TOTAL COST ALL LABOR/LITRE OF ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
127  FORMAT ('0', 1X, 'TOTAL COST ALL LABOR LESS INNOCULATN',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
128  FORMAT ('0', '*TOTAL HOURS OF ALL LABOR TYPES',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
129  FORMAT ('0', 1X, 'TOTAL HRS ALL LABOR USED/LITRE ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
130  FORMAT ('0', 1X, 'TTL HRS ALL LABOR LESS INNOCULATION',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))

131  FORMAT ('0', '*TOTAL COST OF ALL MATERIALS',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
132  FORMAT ('0', 1X, 'TTL COST ALL MATERIALS/LITRE OF ALGAE',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))
133  FORMAT ('0', 1X, 'TTL COST ALL MATER. LESS INNOCULATION',
    B           T41, 6(G13.4,2X)/ ' ', T41, 6(G13.4,2X))


      RETURN
      END




      SUBROUTINE PRICE6 (WRT,RED,HREAD,RUN2,RUN3)
      RETURN
      END







        SUBROUTINE BODY
C       ---------------
```

```
C  ORGNSM:
C  -------
C  SUBROUTINE DOES NOT VARY WITH ORGANISM
C  VARIABLES DO NOT CHANGE

C  STRUCTURE: OPTION TO RUN FOR LESS THAN FULL PRODUCTION SEASON
C             LOOPS THROUGH EACH MONTH OF PRODUCTION
C                LOOPS THROUGH EACH STAGE
C                   DETERMINES PERIODS IN EACH STAGE
C                   LOOPS THROUGH EACH PERIOD OF EACH STAGE
C                      UPDATES CURRENT DAYS AND MONTH
C                      CALLS NUMBER
C                      CALLS INPUT QUANTITY SUBROUTINES (QUANTN)
C                      CALLS COST (COSTJ)
C                   CALLS SUBROUTINE TO OUTPUT (WRTOUT, WOUTO)

C 1. PARAMETERS
      INCLUDE  (IPAR1)
C 2. ARGUMENTS
      INTEGER  INOW, JNOW, LASTJ, MNOW
      REAL   DAYNOW, WKSNOW
      CHARACTER * 10  MBEGUN
C 3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICBLK1)
      INCLUDE (ICBLKO)
      INCLUDE (ICUNIT)
C 4. LOCAL VARIABLES
      INTEGER  ADDTOM,  I, J, M, T
      CHARACTER * 1  QUEST

C  NAME    DESCRIPTION OF LOCAL VARIABLES ARGUMENTS
C  ------  ----------------------------------------------------------
C  ADDTOM  UPDATE MONTH THAT PRODUCTION IS OCCURRING IN
C  DAYNOW  ARGUMENT:  ELAPSED DAYS TO END OF PERIOD
C  I       LOOP COUNTER FOR STAGES
C  INOW    ARGUMENT:  CURRENT STAGE THAT J IS IN
C  J       LOOP COUNTER FOR PERIODS
C  JNOW    ARGUMENT:  CURRENT PERIOD COMPUTING FOR
C  LASTJ   ARGUMENT:  LAST J THAT CALCULATIONS ACCOMPLISHED
C  M       LOOP COUNTER FOR MONTHS
C  MBEGUN  MONTH THAT BATCH IS BEGUN FOR CURRENT CALCULATIONS
C  MNOW    ARGUMENT:  CURRENT MONTH THAT J IS OCCURRING IN
C  QUEST   RUN FOR LESS THAN THE FULL PRODUCTION SEASON?
C  T       LOOP COUNTER FOR TYPES
C  WKSNOW  ARGUMENT:  ELAPSED WEEKS AT END OF J
C  ------------------------------------------------------------------


C  OPTION TO RUN PROGRAM FOR LESS MONTHS THAN FULL PRODUCTION SEASON
      WRITE(WI,20000)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y') THEN
         WRITE(WI,21000)
         READ(RI,*) M1,M2
      ELSE
         M1 = BEGINM
         M2 = ENDM
      ENDIF

1000  FORMAT (A1)

20000 FORMAT(' '///' DO YOU WANT TO RUN THE PROGRAM FOR LESS THAN THE ',
     2          'FULL PRODUCTION SEASON?'/' ','Y OR N')
21000 FORMAT(' '/' TYPE THE NUMBER OF THE MONTH TO BEGIN COMPUTATION ',
     2          'AND THE MONTH TO END'/' ','TYPE TWO INTEGERS')



C  COMPUTE FOR A NEW BATCH ON THE BEGINNING OF EACH MONTH
      DO 100 M=M1,M2

C        NAME THE MONTH THAT PRODUCTION BEGINS
         MBEGUN = NAMEM(M)

C        LOOP THROUGH STAGES
         DAYNOW = 0.
         DO 200 I=1,NOOFI
            INOW = I

C           DEFINE ALL PERIODS SEQUENTIALLY THROUGH ALL STAGES

C           OPTION 1: SKIP ALL I BEFORE 1ST STAGE OF INTEREST - 1
            IF (OPTION .EQ. 1  .AND.  I .LT. (BEGINI-1))  THEN
               BEGINJ(I) = 0
               ENDJ(I)   = 0
               JPERI(I)  = 0
               DAYSJ(I)  = 0.
               GOTO 200

C           OPTION 1:  DEFINE 1ST J TO BE STAGE BEFORE 1ST STG. OF INTER
            ELSEIF (OPTION .EQ. 1  .AND.  I .EQ. (BEGINI-1))  THEN
               BEGINJ(I) = 1
               ENDJ(I)   = 1
               JPERI(I)  = 1
               DAYSJ(I)  = DAYOP1

C           OPTION 2,3:  DEFINE 1ST PERIOD TO BE 1ST PERIOD OF I=1
            ELSEIF (OPTION .NE. 1  .AND.  I .EQ. BEGINI)  THEN
               BEGINJ(I) = 1
               ENDJ(I)   = BEGINJ(I) + JPERI(I) - 1
               DAYSJ(I)  = DAYSI(I) / JPERI(I)

C           OPTIONS ALL:  DEFINE ALL SUBSEQUENT PERIODS AFTER THE 1ST
C           PERIOD AS ACCUMULATING THROUGH THE STAGES
            ELSE
               BEGINJ(I) = ENDJ(I-1) + 1
               ENDJ(I)   = BEGINJ(I) + JPERI(I) -1
               DAYSJ(I)  = DAYSI(I) / JPERI(I)
            ENDIF


C           LOOP THROUGH EACH PERIOD, BY STAGE
            DO 300 J=BEGINJ(I),ENDJ(I)
               JNOW = J
               LASTJ = J

C              ACCUMULATE DAYS AND INCREMENT TIME VARIABLES
               DAYNOW = DAYNOW + DAYSJ(I)
               WKSNOW = DAYNOW / 7.
               ADDTOM = IFIX(DAYNOW-1) / 30
               MNOW   = M + ADDTOM

C              ALLOW FOR FULL YEAR CYCLE
C              (USE THE FACT THAT M IS AN INTEGER TO SUBTRACT PROPERLY)
               IF (ENDM-BEGINM .EQ. 11) MNOW= MNOW- (12* ((MNOW-1) /12))
```

```
            MOFJ  (J) = MNOW
            IOFJ  (J) = I
            DAYUPJ(J) = DAYNOW
            WKSUPJ = WKSNOW

C           HALT IF BEYOND MONTH OF PRODUCTION
            IF (MNOW .GT. ENDM)  THEN
               PASEDM(J) = 1
               GOTO 250

C           COMPUTE QUANTITY OF N, COSTS, AND NUMBER OF ANIMALS
            ELSE
               PASEDM(J) = 0

C              COMPUTE NUMBER AND SIZE OF ORGANISMS IN J
               CALL NUMBRB(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
               CALL SIZE  (INOW,JNOW,WKSNOW,MNOW,DAYNOW)

C              COMPUTE QUANTITY OF EACH INPUT,N, USED IN J
               IF (I .GE. BEGINI) THEN
                 DO 400 N=1,NOOFN
C                   CALL SUBR. TO COMPUTE QUANTITY IF CHOSEN
                    IF (OPNNEW(I,N) .EQ. 'C')  THEN
                     IF(N.EQ.1)CALLQUANT1(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
                     IF(N.EQ.2)CALLQUANT2(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
                     IF(N.EQ.3)CALLQUANT3(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
                     IF(N.EQ.4)CALLQUANT4(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
                     IF(N.EQ.5)CALLQUANT5(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
                     IF(N.EQ.6)CALLQUANT6(INOW,JNOW,WKSNOW,MNOW,DAYNOW)

C                    QUANTITY ENTERED, THEN QUANT. IS SAME EACH PERIOD
                     ELSEIF (OPNNEW(I,N) .EQ. 'E') THEN
                       DO 500 T=1,NOOFT(N)
                          TQUANJ(J,N,T)= TQUANI(I,N,T,MOFJ(J))/JPERI(I)
500                    CONTINUE
                     ENDIF
400              CONTINUE

              ENDIF

C             COMPUTE COST OF INPUTS IN EACH J
              CALL COSTJ(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
           ENDIF

300      CONTINUE

200    CONTINUE
250    CONTINUE

       CALL WRTOUT(MBEGUN,LASTJ,WO)
       IF (QOUTQN(1) .EQ. 'Y')  CALL WOUTN1(MBEGUN,LASTJ,WON(1))
       IF (QOUTQN(2) .EQ. 'Y')  CALL WOUTN2(MBEGUN,LASTJ,WON(2))
       IF (QOUTQN(3) .EQ. 'Y')  CALL WOUTN3(MBEGUN,LASTJ,WON(3))
       IF (QOUTQN(4) .EQ. 'Y')  CALL WOUTN4(MBEGUN,LASTJ,WON(4))
       IF (QOUTQN(5) .EQ. 'Y')  CALL WOUTN5(MBEGUN,LASTJ,WON(5))
       IF (QOUTQN(6) .EQ. 'Y')  CALL WOUTN6(MBEGUN,LASTJ,WON(6))

100   CONTINUE

      WRITE(WI,30000)

C  OPTION TO SEE OUTPUT GENERATED FOR LAST BATCH
      WRITE(WI,40000)
      READ (RI,1000) QUEST
      IF (QUEST .EQ. 'Y') THEN
         CALL WRTOUT(MBEGUN,LASTJ,WI)
         IF (QOUTQN(1) .EQ. 'Y')  CALL WOUTN1(MBEGUN,LASTJ,WI)
         IF (QOUTQN(2) .EQ. 'Y')  CALL WOUTN2(MBEGUN,LASTJ,WI)
         IF (QOUTQN(3) .EQ. 'Y')  CALL WOUTN3(MBEGUN,LASTJ,WI)
         IF (QOUTQN(4) .EQ. 'Y')  CALL WOUTN4(MBEGUN,LASTJ,WI)
         IF (QOUTQN(5) .EQ. 'Y')  CALL WOUTN5(MBEGUN,LASTJ,WI)
         IF (QOUTQN(6) .EQ. 'Y')  CALL WOUTN6(MBEGUN,LASTJ,WI)
      ENDIF

30000 FORMAT(' '//' ****** COMPUTATIONS COMPLETED ******'/)

40000 FORMAT(' '/' DO YOU WANT TO SEE THE COMPUTATIONS FOR THE LAST ',
     2       'BATCH?')


      RETURN
      END




      SUBROUTINE NUMBRB(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
C     ----------------------------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES MUST BE RETURNED:
C    NUMBER OR ORGANISMS IN J:      ONOOFJ(J)
C    NUMBER OF ORGANISM UP TO J:    ONOUPJ(J)
C    AGE OF ORGANISM UP TO END J:   AGEUPJ(J)
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE: COMPUTES AND RETURNS THE NUMBER OF ORGANISMS REMAINING AT
C             THE END OF EACH PERIOD, ONOUPJ, AND THE NUMBER OF AVERAGE
C             ORGANISMS DURING EACH PERIOD, ONOOFJ

C  1. PARAMETERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS AND FUNCTIONS
      INTEGER  INOW, JNOW, MNOW
      REAL     AGENO, WKSNOW, DAYNOW
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICORG)
      INCLUDE (ICBLKO)
C  4. LOCAL VARIABLES
      INTEGER  D, DCNTR, I, J, M, VALUE
      REAL     DAY, DWK, TNO, WKS

C  NAME    DESCRIPTION OF LOCAL VARIABLES
C  ------  -------------------------------------------------------------
C  AGENO   FUNCTION: RELATES AGE/SURVIVAL, CALLED EACH DAY
C  D       LOOP COUNTER FOR DAYS
C  DAY     TRANSLATE ARGUMENT TO LOCAL DAYS
C  DAYNOW  ARGUE:  ELAPSED DAYS TO END OF CURRENT J
C  DCNTR   NO. OF DAYS LOOPED THROUGH IN PERIOD
C  DWK     TRANSLATE DAY LOOP INTO WEEKS
C  I       TRANSLATE INOW INTO CURRENT STAGE
```

```
C  INOW    ARGUE:  STAGE THAT CURRENT PERIOD IS IN
C  J       TRANSLATES JNOW FOR CURRENT PERIOD
C  JNOW    ARGUE:  CURRENT PERIOD BEING CALCULATED
C  M       TRANSLATE MNOW FOR CURRENT MONTH
C  MNOW    ARGUE:  MONTH THAT CURRENT J IS IN
C  TNO     SUM OF ALL ORGANISMS EXISTING EACH DAY OF PERIOD
C  VALUE   VALUE TO DECREASE REIMANN SUM.  1ST TERM OF SUM
C  WKS     TRANSLATE WKSNOW FOR ELAPSED WEEKS
C  WKSNOW  ARGUE:  ELAPSED WEEKS FOR CURRENT J
C  ------------------------------------------------------------------

C  TRANSLATE LOCAL VARIABLES
      I   = INOW
      J   = JNOW
      WKS = WKSNOW
      M   = MNOW
      DAY = DAYNOW


C  ALGORITHM DEPENDS ON TYPE OF ORGANISM

      IF (OTYPE .EQ. 1  .OR.  OTYPE .EQ. 2)  THEN
C        WHEN I=CONDITIONING, ASSUME NO DEATH RATE AND ANIMALS ALL AT
C        TWO YEARS OLD SO AS NOT TO EXCEED MAXIMUM OF AGE/SZE EQUATION
         IF (I .EQ. 1)  THEN
            AGEUPJ(J)  =  365. * 2.
            ONOOFJ(J)  =  NOOI1
            ONOUPJ(J)  =  NOOI1

C        ALL OTHER STAGES, COMPUTE NUMBER OF ORGANISMS SURVIVING AT EACH
C        DAY OF THE PERIOD BY CALLING FUNCTION AGENO.  THEN FIND AVERAGE
C        SURVIVAL OF THE PERIOD BY SUMMING USING THE TRAPEZOIDAL RULE
         ELSEIF (I .GT. 1)  THEN

            DCNTR = 0
            TNO = 0.
            DO 100 D = (DAY-DAYSJ(I)-DAYSI1+1.), (DAY-DAYSI1+.49999)
               DWK = FLOAT(D) / 7.
               ONOUPJ(J) = AGENO(DWK)
               TNO = TNO + ONOUPJ(J)
               DCNTR = DCNTR + 1
100         CONTINUE

C           COMPUTE NUMBER DURING DAY BEFORE PERIOD BEGINS
            DWK = (DAY-DAYSJ(I)-DAYSI1) / 7.
            VALUE = AGENO(DWK)

C           INCREASE TOTAL BY 1/2 VALUE, DECREASE BY 1/2 LAST SUM TERM
            TNO  =  TNO + ((VALUE - ONOUPJ(J)) / 2.)

C           COMPUTE AVERAGE VALUE WHERE DELTA T IS ALWAYS = 1
            ONOOFJ(J) = TNO / FLOAT(DCNTR)

C           AGE OF ANIMAL AT END OF PERIOD DOESNT INCLUDE CONDITIONING
            AGEUPJ(J)  =  DAY - DAYSI1
         ENDIF

      ELSE
         STOP 'ORGANISM TYPE INCORRECT IN SUBR. NUMBRB'
      ENDIF


      RETURN
      END






      SUBROUTINE SIZE(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
C     ---------------------------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES THAT MUST BE RETURNED TO  ORGNSM:
C    SIZE AT END OF PERIOD  :  SZEUPJ(J)
C    AVERAGE SIZE IN PERIOD :  SZEOFJ(J)
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  COMPUTES AND RETURNS SIZE OF ORGANISM DURING EACH PERIOD
C              AND AT END OF PERIOD USING THE FUNCTION AGESZE

C  VARIABLE DECLARATIONS:
C  1. PARAMETERS
      INCLUDE (IPAR1)
C  2. ARGUMENTS AND FUNCTIONS
      INTEGER INOW, JNOW, MNOW
      REAL    DAYNOW, AGESZE, WKSNOW
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICORG)
      INCLUDE (ICBLKO)
C  4. LOCAL VARIABLES
      INTEGER D, DCNTR, I, J, M, VALUE
      REAL  DAY, DWK, TSZE, WKS

C  NAME   DESCRIPTION OF LOCAL AND ARGUMENT VARIABLES
C  ------ -----------------------------------------------------------
C  AGESZE FUNCTION: RELATES AGE/SIZE. CALLED EACH DAY OF PERIOD
C  D      LOOP COUNTER FOR DAYS
C  DAY    TRANSLATE DAYNOW FOR ELAPSED DAYS OF PERIOD
C  DAYNOW ARGUE: ELAPSED DAYS TO END OF PERIOD
C  DCNTR  LAST DAY OF PERIOD LOOPED THROUGH
C  DWK    TRANSLATE DAYS LOOPED IN PERIOD INTO WEEKS
C  I      TRANSLATE INOW INTO STAGE OF CURRENT PERIOD
C  INOW   ARGUE:  STAGE THAT CURRENT J IS IN
C  J      TRANSLATE JNOW FOR CURRENT PERIOD
C  JNOW   ARGUE:  CURRENT PERIOD
C  M      TRANSLATE MNOW FOR MONTH THAT CURRENT PERIOD OCCURS IN
C  MNOW   ARGUE:  MONTH THAT CURRENT PERIOD OCCURS IN
C  TSZE   SUM OF SIZE FOR ALL DAYS OVER THE PERIOD
C  VALUE  VALUE OF 1ST TERM TO SUBTRACT OFF OF THE REIMANN SUM
C  WKS    TRANSLATE WKSNOW FOR ELAPSED WEEKS
C  WKSNOW ARGUE: ELAPSED WEEKS TO END OF CURRENT PERIOD
C  ------------------------------------------------------------------


C  TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
      I   = INOW
      J   = JNOW
      WKS = WKSNOW
      M   = MNOW
      DAY = DAYNOW
```

```
C  ALGORITHM DEPENDS ON TYPE OF ORGANISM
      IF (OTYPE .EQ. 1  .OR.  OTYPE .EQ. 2)  THEN

C        CONDITIONING STAGE, ASSUME SIZE CONSTANT AND ALL AT 3 YEARS
         IF (I .EQ. 1)  THEN
            DWK  =  AGEUPJ(J) / 7.
            SZEUPJ(J) = AGESZE(DWK)
            SZEOFJ(J) = SZEUPJ(J)

C        OTHERWISE, COMPUTE SIZE AT EACH DAY OF PERIOD BY CALLING
C        FUNCTION AGESZE, THEN FIND AVG. SIZE OF PERIOD BY SUMMING
C        USING THE TRAPEZOIDAL RULE
         ELSEIF (I .GT. 1) THEN

            DCNTR = 0
            TSZE = 0.
            DO 100 D = (DAY-DAYSJ(I)-DAYSI1+1.), (DAY-DAYSI1+.4999)
               DWK = FLOAT(D) / 7.
               SZEUPJ(J) = AGESZE(DWK)
               TSZE = TSZE + SZEUPJ(J)
               DCNTR = DCNTR + 1
100         CONTINUE

C           COMPUTE SIZE DURING DAY BEFORE PERIOD BEGINS
            DWK = (DAY-DAYSJ(I)-DAYSI1) / 7.
            VALUE = AGESZE(DWK)

C           INCREASE TOTAL BY 1/2 VALUE, DECREASE BY 1/2 LAST SUM TERM
            TSZE  =  TSZE + ((VALUE - SZEUPJ(J)) / 2.)

C           COMPUTE AVERAGE SIZE DURING J, WHERE DELTA T ALWAYS = 1
            SZEOFJ(J) = TSZE / FLOAT(DCNTR)
         ENDIF

      ELSE
         STOP 'ORGANISM TYPE WRONG IN SUBROUTINE SIZE'
      ENDIF


      RETURN
      END






      SUBROUTINE QUANT1(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
      RETURN
      END




      SUBROUTINE QUANT2(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
      RETURN
      END




      SUBROUTINE QUANT3(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
      RETURN
      END




      SUBROUTINE QUANT4(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
      RETURN
      END




      SUBROUTINE QUANT5(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
C     ----------------------------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES RETURNED (ONLY IF OPTION TO COMPUTE Q FOR STAGES CHOSEN):
C  QUANT OF INPUT 5, IN J, OF TYPE T, CURRENT TIME:  TQUANJ(J,5,T)
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  COMPUTES THE QUANTITY OF EACH TYPE,T, OF INPUT 5,
C              USED IN EACH PERIOD IN ANY PARTICULAR M
C              ASSUMES THAT NO ALGAE IS PURCHASED FROM OUTSIDE SOURCES

C  1. PARAMETERS
      INCLUDE (IPAR1)
C  2. ARGUMENTS AND FUNCTIONS
      INTEGER  INOW, JNOW, MNOW
      REAL  AGENO, AGESZE, WKSNOW, DAYNOW
C  3. COMMON BLOCKS
      INCLUDE (ICBLK1)
      INCLUDE (ICBLKO)
      INCLUDE (ICMAIN)
      INCLUDE (ICORG)
C  4. LOCAL VARIABLES
      INTEGER  D, DCNTR, I, J, M, T
      REAL     CELLS, CDWK, DAY, DWK, FCELLS, NODWK, SZEDWK, TCELLS,
     R         TCDWK, WKS

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------  -----------------------------------------------------
C  AGENO   FUNCTION: RELATES AGE/SURVIVAL. CALLED EACH DAY OF PERIOD
C  AGESZE  FUNCTION: RELATES AGE/SIZE. CALLED EACH DAY OF PERIOD
C  CELLS   CELLS CONSUMED GIVEN FILTERING RATE AND REQ.
C  CDWK    CELLS(10,000) ALGAE EATEN/BIVALVE EACH DAY OF PERIOD
C  D       LOOP COUNTER FOR DAYS
C  DAY     TRANSLATE DAYNOW FOR ELAPSED DAYS
C  DAYNOW  ARGUE:  ELAPSED DAYS TO END OF PERIOD
C  DCNTR   USED TO KEEP TRACK OF NUMBER OF DAYS LOOPED THRU A PERIOD
C  DWK     FOR EACH DAY OF PERIOD, TRANSLATE INTO ELAPSED WEEKS
C  FCELLS  CELLS REQUIRED CONSIDERING FILTERING RATE/STAGE
C  I       TRANSLATE INOW FOR CURRENT STAGE
C  INOW    ARGUE:  CURRENT STAGE THAT J IS IN
C  J       TRANSLATE JNOW FOR CURRENT PERIOD
C  JNOW    ARGUE: CURRENT PERIOD
C  M       TRANSLATE MNOW FOR CURRENT MONTH
C  MNOW    ARGUE:  MONTH THAT CURRENT PERIOD IS IN
C  NODWK   NUMBER OF BIVALVE. EXISTING AT EACH DAY OF PERIOD
C  SZEDWK  SIZE OF BIVALVE AT EACH DAY OF PERIOD
C  T       LOOP COUNTER FOR TYPES OF INPUT N
C  TCELLS  SUM OF TCDWK OVER ENTIRE PERIOD
C  TCDWK   CELLS EATEN FOR ALL BIVALVES AT THAT DAY OF PERIOD
C  WKS     TRANSLATE WKSNOW FOR ELAPSED WEEKS
```

```
C  WKSNOW  ARGUE:  ELAPSED WEEKS TO END OF PERIOD
C  ----------------------------------------------------------------

C TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
      I   = INOW
      J   = JNOW
      WKS = WKSNOW
      M   = MNOW
      DAY = DAYNOW

C  ALGORITHM USED SHOULD DEPEND ON ORGANISM TYPE
      IF (OTYPE .EQ. 1   .OR.  OTYPE .EQ. 2)  THEN

C        CONDITIONING STAGE REQUIRES DIFFERENT ALGORITHM
         IF (I .EQ. 1)  THEN
            STOP 'ERROR SUBR. QUANT5.  CANT COMPUTE N5 FOR I=1'

C        OTHERWISE, COMPUTE TOTAL CELLS USED OVER PERIOD BY SUMMING
C        OVER THE PERIOD THE AMOUNT REQUIRED EACH DAY OF THE PERIOD.
C        USE THE TRAPEZOIDAL RULE
         ELSEIF (I .GT. 1)  THEN

            DCNTR  = 0
            TCELLS = 0.
            DO 100 D = (DAY-DAYSJ(I)-DAYSI1+1.), (DAY-DAYSI1+.499)
               DWK = FLOAT(D) / 7.

C              USE THE ALGORITHM CHOSEN IN READN5 TO FIND CELLS/BIVALV
               IF (QEQQ5 .EQ. 1)  THEN
                  SZEDWK = AGESZE(DWK)
                  CDWK = ((SZEDWK) **(1/EQQ5B(1))) *
     2                   (EXP ((-EQQ5A(1)) /EQQ5B(1)) )
               ELSEIF (QEQQ5 .EQ. 2)  THEN
                  CDWK = ((DWK) **(1/EQQ5B(2))) *
     2                   (EXP ((-EQQ5A(2)) / EQQ5B(2)) )
               ENDIF

C              TOTAL CELLS/DAY IS FUNCTION OF NUMBER OF ANIMALS AT THAT
C              DAY.  ALSO MULTIPLY BY 10000. FOR PROPER EQUATION
               NODWK = AGENO(DWK)
               TCDWK = CDWK * NODWK * 10000.
               TCELLS = TCELLS + TCDWK
               DCNTR = DCNTR + 1

100         CONTINUE

C           DETERMINE QUANTITY OF DAY BEFORE PERIOD BEGINS, TO MORE
C           CLOSELY APPROXIMATE THE INTEGRAL
            DWK = (DAY-DAYSJ(I)-DAYSI1) / 7.
            IF (QEQQ5 .EQ. 1) THEN
               SZEDWK = AGESZE(DWK)
               VALUE = ((SZEDWK) **(1/EQQ5B(1))) *
     2                (EXP((-EQQ5A(1) / EQQ5B(1))))
            ELSEIF (QEQQ5 .EQ. 2) THEN
               VALUE = ((DWK) **(1/EQQ5B(2))) *
     2                (EXP((-EQQ5A(2) / EQQ5B(2))))
            ENDIF
            NODWK = AGENO(DWK)
            VALUE = VALUE * NODWK * 10000.

C           INCREASE SUM BY 1/2 VALUE, DECREASE BY 1/2 LAST SUM TERM
            TCELLS  =  TCELLS + ((VALUE - TCDWK) / 2.)

C           FILTERING RATE_100%, THEN INCREASE AMOUNT/PERIOD
            FCELLS = TCELLS/FILTER(I)

            DO 200 T=1,NOOFT(5)

C              COMPUTE CELLS EACH TYPE, BASED ON 100% USED/STAGE
               CELLS = FCELLS * N5REQ(I,T)

C              COMPUTE QUANT/TYPE MUST PAY FOR SINCE IS REDUCED BY QUANT
C              AVAIL. FROM SEAWATER.  ASSUME SAME % EACH TYPE IN THE SEA
               TQUANJ(J,5,T)  =  CELLS - (CELLS * SEACEL(I,M))

C              COMPUTE AVERAGE QUANTITY USED/DAY OF EACH TYPE, TO BE
C              PRINTED OUT IN ADDITIONAL INPUT 5 SUBROUTINE, WOUTQN(5)
               QUAOFJ(J,T) = TQUANJ(J,5,T) / FLOAT(DCNTR)

200         CONTINUE

         ENDIF
      ELSE
         STOP 'ORGANISM TYPE INCORRECT IN SUBR. QUANT5'
      ENDIF



      RETURN
      END




      SUBROUTINE QUANT6(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
      RETURN
      END







      SUBROUTINE COSTJ(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
C     ----------------------------------------------

C  ORGNSM:
C  ------
C  SUBROUTINE DOES NOT VARY WITH ORGANISM
C  VARIABLES DO NOT CHANGE

C  STRUCTURE: COMPUTES COSTS OF PERIOD J
C             COSTS ACCUMULATED IN J
C             COSTS ACCUMULATED UP TO THE END OF J
C             COSTS PER ORGANISM UP TO THE END OF J

C  1. PARAMETERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS
      INTEGER  INOW, JNOW, MNOW
      REAL     WKSNOW, DAYNOW
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICBLK1)
      INCLUDE (ICBLKO)
      INCLUDE (ICUNIT)
C  4. LOCAL VARIABLES
```

```
      INTEGER  I, J, M, N, T
      REAL     WKS, DAY, CSTOFJ

C  NAME    DESCRIPTION OF LOCAL VARIAB AND ARGUMENTS
C  ------  ------------------------------------------------------------
C  CSTOFJ COST OF TYPE T IN J (WRITTEN OVER)
C  DAY    TRANSLATE DAYNOW FOR ELAPSED DAYS
C  DAYNOW ARGUE:  ELAPSED DAYS TO END OF PERIOD
C  I      TRANSLATE INOW FOR CURRENT STAGE
C  INOW   ARGUE:  CURRENT STAGE THAT J IS IN
C  J      TRANSLATE JNOW FOR CURRENT PERIOD
C  JNOW   ARGUE:  CURRENT PERIOD
C  M      TRANSLATE MNOW FOR CURRENT MONTH
C  MNOW   ARGUE:  MONTH THAT CURRENT PERIOD IS IN
C  N      LOOP COUNTER FOR INPUTS
C  T      LOOP CONTER FOR TYPES
C  WKS    TRANSLATE WKSNOW FOR ELAPSED WEEKS
C  WKSNOW ARGUE:  ELAPSED WEEKS TO END OF PERIOD
C  --------------------------------------------------------------------


C  TRANSLATE ARGUMENTS TO LOCAL VARIABLE
      I = INOW
      J = JNOW
      WKS = WKSNOW
      M = MNOW
      DAY = DAYNOW

C  COMPUTE COST OF EACH INPUT, N, ONE AT A TIME
      CSUPJ(J) = 0.
      CSOFJ(J) = 0.
      DO 100 N=1,NOOFN

         IF (OPTION .EQ. 1  .AND.  J .EQ. 1)  THEN
            CSOFJ(J) = CSOP1
            CSUPJ(J) = CSOP1
            CSNOFJ(J,N) = 0.
            CSNUPJ(J,N) = 0.
         ENDIF

C        COST OF EACH TYPE,T, OF EACH INPUT,N, DURING A PERIOD,J
         IF (I .GE. BEGINI)  THEN
           CSNOFJ(J,N) = 0.
           DO 200 T=1,NOOFT(N)
             CSTOFJ = TQUANJ(J,N,T) * PRICET(N,T,MOFJ(J))
             CSNOFJ(J,N) = CSNOFJ(J,N) + CSTOFJ
200        CONTINUE

C  IE, LET THIS VARIABLE EQUAL CALCULATIONS FOR LAST DAY
C          COST OF ALL INPUTS USED IN A PERIOD
           CSOFJ(J) = CSOFJ(J) + CSNOFJ(J,N)

           IF (J .EQ. 1 )  THEN
              CSNUPJ(J,N) = CSNOFJ(J,N)
              CSUPJ(J)    = CSOFJ(J)
           ELSEIF (J .NE. 1)  THEN
              CSNUPJ(J,N) = CSNOFJ(J,N) + CSNUPJ(J-1,N)
              CSUPJ(J) = CSOFJ(J) + CSUPJ (J-1)
           ENDIF
         ENDIF

100   CONTINUE

C  COMPUTE COST PER ORGANISM
C     AVERAGE PER DAY OVER THE PERIOD
      CSOOFJ(J)  =  (CSOFJ(J) / ONOOFJ(J))  /  DAYSJ(IOFJ(J))
C     UP TO END OF PERIOD
      IF (ONOUPJ(J) .EQ. 0) THEN
         CSOUPJ(J) = 0.
      ELSE
         CSOUPJ(J) = CSUPJ(J) / ONOUPJ(J)
      ENDIF


      RETURN
      END







      SUBROUTINE WRTOUT(MBEGUN,LASTJ,WRT)
C     ---------------------------------

C  ORGNSM:
C  -------
C  SUBROUTINE DOES NOT VARY WITH ORGANISM
C  VARIABLES DO NOT CHANGE

C  STRUCTURE:  WRITES OUT INFORMATION COMPUTED FOR A BATCH BEGUN ON THE
C                FIRST OF EACH MONTH OF THE PRODUCTION SEASON.
C              THIS INFORMATION INCLUDES:
C               -STAGE AND PERIOD INFORMATION
C               -QUANTITY OF EACH TYPE OF EACH INPUT USED IN EACH PERIOD
C               -COST OF EACH INPUT USED DURING EACH PERIOD AND UP TO A
C                PERIOD
C               -COST OF PRODUCING AN ORGANISM UP TO THE END OF A PERIOD

C  1. PARAMETERS
      INCLUDE  (IPAR1)
C  2. ARGUMENTS
      INTEGER  LASTJ, WRT, WRT2
      CHARACTER * 10  MBEGUN
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICBLK1)
      INCLUDE (ICBLKO)
      INCLUDE (ICUNIT)
C  4. LOCAL VARIABLES
      INTEGER  I, J, M, N, T, W

C  NAME    DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------  ------------------------------------------------------------
C  I       LOOP COUNTER FOR STAGE
C  J       LOOP COUNTER FOR PERIODS
C  LASTJ   ARGUE:  LAST J PERIOD SUCCESSFULLY COMPUTED
C  M       LOOP COUNTER FOR MONTHS
C  MBEGUN  MONTH THAT CURRENT BATCH WAS BEGUN
C  N       LOOP COUNTER FOR INPUTS
C  T       LOOP COUNTER FOR TYPES
C  WRT     ARGUMENT:  UNIT NUMBER
C  W       TRANSLATE ARGUMENT UNIT NUMBER TO LOCAL WRITE VARIABLE
C  WRT2    ARGUE:  TRANSLATE UNIT WRITE TO NEXT LEVEL SUBROUTINE
C  -------------------------------------------------------------------
```

```
      W = WRT
      WRT2 = WRT


C  FIRST PAGE INFORMATION FOR OUTPUT FILE
      IF (MBEGUN .EQ. NAMEM(M1))  THEN
         WRITE(W,10000) NAME, DATE, NAMEF1, NAME, NAMEO(OTYPE), NAME,
     2                  ONOBEG, NAMES, OPTION, NAMEI(BEGINI)
         IF (OPTION .EQ. 1)  WRITE(W,13000) BEGINI-1

C        PRICE MATRIX FOR EACH INPUT
         WRITE(W,16000) (NAMEM(M), M=BEGINM,ENDM)
         DO 100 N=1,NOOFN
            IF (NOOFT(N) .EQ. 1) THEN
               WRITE(W,16100) NAMEN(N), NAMENU(N),
     2                        (PRICET(N,1,M), M=BEGINM,ENDM)
            ELSEIF (NOOFT(N) .GT. 1)  THEN
               DO 150 T=1,NOOFT(N)
                  WRITE(W,16100) NAMET(N,T), NAMENU(N),
     2                           (PRICET(N,T,M), M=BEGINM,ENDM)
150            CONTINUE
            ENDIF
100      CONTINUE
      ENDIF


C  HEADINGS FOR EACH NEW MONTH BEGUN
      WRITE(W,20000) NAME, DATE, MBEGUN

C  AGE, SIZE AND TIME
      WRITE(W,32000) NAME, NAME, NAME
      WRITE(W,90000)
      DO 300 J=1,LASTJ
         IF (PASEDM(J) .EQ. 0 ) THEN
           WRITE(W,33000)  J, IOFJ(J),NAMEI(IOFJ(J)),MOFJ(J),DAYUPJ(J),
     2           AGEUPJ(J), ONOOFJ(J), ONOUPJ(J), SZEOFJ(J), SZEUPJ(J)
         ELSEIF (PASEDM(J) .EQ. 1) THEN
            WRITE(W,33100) J, IOFJ(J), NAMEI(IOFJ(J)), MOFJ(J),
     2                     DAYUPJ(J), NAME, MBEGUN
         ENDIF
300   CONTINUE

C  TOTAL COSTS
      WRITE(W,35000) NAME, NAME
      WRITE(W,93000)
      DO 350 J=1,LASTJ
         IF (PASEDM(J) .EQ. 0) THEN
            WRITE(W,36000) J, IOFJ(J), CSOFJ(J), CSUPJ(J), CSOOFJ(J),
     2                     CSOUPJ(J)
         ELSEIF (PASEDM(J) .EQ.1) THEN
            WRITE(W,42100) J, IOFJ(J)
         ENDIF
350   CONTINUE

C  COSTS IN A PERIOD, J OF EACH INPUT
      WRITE(W,40000)
      WRITE(W,41000) (NAMEN(N), N=1,NOOFN)
      WRITE(W,95000)
      DO 400 J = BEGINJ(BEGINI), LASTJ
         IF (PASEDM(J) .EQ. 0 ) THEN
            WRITE(W,42000) J, IOFJ(J), (CSNOFJ(J,N), N=1,NOOFN)
         ELSEIF (PASEDM(J) .EQ. 1) THEN
            WRITE(W,42100)  J, IOFJ(J)
         ENDIF
400   CONTINUE

C  COST OF INPUTS UP TO THE END OF A PERIOD, J
      WRITE(W,50000)
      WRITE(W,41000) (NAMEN(N), N=1,NOOFN)
      WRITE(W,95000)
      DO 500 J = BEGINJ(BEGINI), LASTJ
         IF (PASEDM(J) .EQ. 0)  THEN
            WRITE(W,42000)  J, IOFJ(J), (CSNUPJ(J,N), N=1,NOOFN)
         ELSEIF (PASEDM(J) .EQ. 1)  THEN
            WRITE(W,42100)  J, IOFJ(J)
         ENDIF
500   CONTINUE

C  QUANTITY OF INPUT TYPE, T, USED IN A PERIOD, J
      WRITE(W,70000)
      DO 700 N=1,NOOFN
         WRITE(W,71000) NAMEN(N), NAMENU(N)
         IF (NOOFT(N) .GT. 1) THEN
            WRITE(W,72000) (NAMET(N,T), T=1,NOOFT(N))
         ELSE
            WRITE(W,72100)
         ENDIF
         DO 710 J=BEGINJ(BEGINI),LASTJ
          IF (PASEDM(J) .EQ. 0) THEN
           WRITE(W,73000)J,IOFJ(J),(TQUANJ(J,N,T),T=1,NOOFT(N))
          ELSEIF (PASEDM(J) .EQ. 1) THEN
           WRITE(W,42100) J, IOFJ(J)
          ENDIF
710      CONTINUE
700   CONTINUE


10000 FORMAT('1',A,' COST SIMULATION MODEL:  OUTPUT GENERATED'/
     2          '+',52('-')/
     3          '0','DATE:  ',A/
     3          ' ','FILE:  ',A///
     4          '0',A,' TYPE:',T24,A/
     5          ' ','NUMBER ',A7,' BEGUN:',T24,G10.5/
     6          ' ','SYSTEM USED: ',T24,A/
     7          ' ','OPTION CHOSEN: ',T24, I1,', FIRST STAGE OF INTEREST',
     8              ' IS THE ',A,' STAGE')

13000 FORMAT(' ',T24,'(*NOTE: STAGE ',I1,' INCLUDES ALL STAGES UP TO ',
     2              'THAT STAGE)')

16000 FORMAT('0'/' ','PRICE OF INPUTS PER MONTH',/'+',27('_')/
     2          '0','MONTH',T19,6(3X,A10,4X)/' ',T19,6(3X,A10,4X))

16100 FORMAT(' ',A10,'/',A5,T19,6(3X,G14.7)/' ',T19,6(3X,G14.7))

20000 FORMAT('1',A,' COST SIMULATION MODEL: OUTPUT GENERATED (CONT.)'/
     2            ' ',A/'0','BATCH BEGUN: ',1X,A/'+',24('_')/)

32000 FORMAT('0',T11,'NAME OF',T31,'TIME UP TO',T46,'AGE AT',T61,'AVG ',
     2     'NO. ',A6,T76,'NO. ',A6,' AT',T91,'AVG SIZE',T106,'SIZE ',A6/
     3         ' ','PRD',T6,'*STG',T11,'STAGE',T26,'MNTH',
     4          ' ',T31,'END OF PERIOD',T46,'END OF PERIOD',T61,'IN  ',
     5     'PERIOD',T76,'END OF PERIOD',T91,'IN PERIOD',T106,'AT END ',
     6     'PERIOD')

35000 FORMAT('0',T16,'COST OF',T31,'COST UP TO END',T46,'AVG COST ',A3,
     2          T61,'COST ',A6/
     3          ' ','PERIOD',T9,'STAGE',T16,'PERIOD',T31,'OF PERIOD',
```

```
    4       T46,'IN PERIOD/DAY',T61,'AT END PERIOD')

36000 FORMAT(' ',I3,T11,I1,T16,G14.7,T31,G14.7,T46,G14.7,T61,G14.7)

33000 FORMAT(' ',I2,T8,I1,T11,A10,T26,I2,T31,G14.7,T46,G14.7,T61,
    2          G14.7,T76,G14.7,T91,G14.7,T106,G14.7)

33100 FORMAT(' ',I2,T8,I1,T11,A10,T26,I2,T31,G14.7,T46,'TIME PAST ',
    2       'MONTHS OF ',
    3       'PRODUCTION. CANT PRODUCE ',A,' OF THIS AGE IF BEGUN IN ',A)

40000 FORMAT('0'/' ','COST OF INPUTS IN A PERIOD:'/'+',27('_'))

41000 FORMAT(' ','PERIOD',T9,'STAGE',T16,6(A10,5X))

42000 FORMAT(' ',I3,T11,I1, 4(T16,7(G14.7,1X)/' '))

42100 FORMAT(' ',I3,T11,I1,T16,'TIME PAST PRODUCTION MONTHS')

50000 FORMAT('0'/' ','COST OF INPUTS UP TO END OF A PERIOD'/'+',37('_'))

70000 FORMAT('0'//' ','QUANTITY OF INPUT USED PER PERIOD:'/'+',34('_'))

71000 FORMAT(' '/' ',A10,1X,A,':'/' ',17('-'))

72000 FORMAT(' ','PERIOD',T9,'STAGE',T16,10(A10,5X))

72100 FORMAT(' ','PERIOD',T9,'STAGE')

73000 FORMAT(' ',I3,T11,I1,T16,10(G14.7,1X))

90000 FORMAT(' ',3('-'),T6,4('-'),T11,14('-'),T26,4('-'),T31,14('-'),
    2    T46,14('-'),T61,14('-'),T76,14('-'),T91,14('-'),T106,14('-'))

93000 FORMAT(' ',6('-'),T9,6('-'),T16,14('-'),T31,14('-'),T46,14('-'),
    2       T61,14('-'))

95000 FORMAT(' ',6('-'),T9,6('-'),T16,14('-'),T31,14('-'),T46,14('-'),
    2       T61,14('-'),T76,14('-'),T91,14('-'))


      RETURN
      END



      SUBROUTINE WOUTN1(MBEGUN,LASTJ,WRT)
      RETURN
      END



      SUBROUTINE WOUTN2(MBEGUN,LASTJ,WRT)
      RETURN
      END



      SUBROUTINE WOUTN3(MBEGUN,LASTJ,WRT)
      RETURN
      END



      SUBROUTINE WOUTN4(MBEGUN,LASTJ,WRT)
      RETURN
      END



      SUBROUTINE WOUTN5(MBEGUN,LASTJ,WRT)
C     ----------------------------------

C  ORGNSM:
C  --------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES CHANGE WITH ORGANISM
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  WRITE INFORMATION INTO A SEPARATE DATA FILE THAT WILL BE
C              USED FOR ADDITIONAL ANALYSIS BY A STATISTICAL PROGRAM
C              NOTE:THE USE OF THIS SUBROUTINE REQUIRES THAT THE
C              NUMBER OF ALGAE STAGES BE SET AT 6

C  1. PARAMETERS
      INCLUDE(IPAR1)
C  2. ARGUMENTS
      INTEGER LASTJ, WRT
      CHARACTER*10  MBEGUN
C  3. COMMON BLOCKS
      INCLUDE (ICUNIT)
      INCLUDE (ICMAIN)
      INCLUDE (ICORG)
C  4. LOCAL VARIABLES
      INTEGER J, W

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ ------------------------------------------------------------
C  J      LOOP COUNTER FOR PERIODS
C  LASTJ  ARGUE:  LAST PERIOD THAT COMPUTATIONS WERE DONE FOR
C  MBEGUN ARGUE:  MONTH IN WHICH CURRENT BATCH IS BEGUN
C  WRT    ARGUE:  UNIT NUMBER FOR WRITING
C  W      UNIT NUMBER TO WRITE
C  -------------------------------------------------------------------


      W = WRT

C  WRITE INFORMATION SPECIFIC TO THIS PROGRAM
      DO 100 J = 1,LASTJ
         WRITE(W,10000)  J, MOFJ(J), DAYUPJ(J), AGEUPJ(J), ONOOFJ(J),
    2                      ONOUPJ(J), SZEOFJ(J), SZEUPJ(J), CSOFJ(J),
    3                      CSUPJ(J), CSOOFJ(J), CSOUPJ(J), CSNOFJ(J,5),
    4                      CSNUPJ(J,5), TQUANJ(J,5,5), TQUANJ(J,5,6),
    5                      QUAOFJ(J,5), QUAOFJ(J,6)
100   CONTINUE


10000 FORMAT('0',I2,1X,I2,T10,8E12.5/' ',T10,8E12.5)
```

```
      RETURN
      END



      SUBROUTINE WOUTN6(MBEGUN,LASTJ,WRT)
      RETURN
      END






      REAL FUNCTION  AGENO(WEEKS)
C     ---------------------------

C  ORGNSM:
C  -------
C  FUNCTION VARIES WITH ORGANISM
C  FUNCTION REQUIRED FOR SUBROUTINE NUMBRB
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  RELATES SIZE TO AGE IN TERMS OF WEEKS

C  1. PARAMETERS
      INCLUDE (IPAR1)
C  2. ARGUMENTS
      REAL  WEEKS
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICBLKO)

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ -------------------------------------------------------------
C  WEEKS  ARGUE: ELAPSED WEEKS UP TO THE END OF J
C  --------------------------------------------------------------------


C  ALGORITHM USED SHOULD DEPEND ON ORGANISM TYPE
      IF (OTYPE .EQ. 1  .OR.  OTYPE .EQ. 2)  THEN
         AGENO  =  ONOBEG * ((WEEKS+1)**(EQNOA))

      ELSE
         STOP  'ORGANISM TYPE WRONG IN FUNCTION AGENO'
      ENDIF


      RETURN
      END






      REAL FUNCTION  AGESZE(WEEKS)
C     ----------------------------

C  ORGNSM:
C  -------
C  FUNCTION VARIES WITH ORGANISM
C  FUNCTION RETURNED TO SUBROUTINES QUANT5, SIZE
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  RELATES SIZE TO AGE IN TERMS OF WEEKS

C  1. PARAMETERS
      INCLUDE (IPAR1)
C  2. ARGUMENTS
      REAL  WEEKS
C  3. COMMON BLOCKS
      INCLUDE (ICMAIN)
      INCLUDE (ICBLKO)

C  NAME   DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C  ------ -------------------------------------------------------------
C  WEEKS  ARGUE: ELAPSED WEEKS UP TO THE END OF PERIOD J
C  --------------------------------------------------------------------


C  ALGORITHM USED SHOULD DEPEND ON ORGANISM TYPE
      IF (OTYPE .EQ. 1  .OR.  OTYPE .EQ. 2)  THEN
         AGESZE  =  (EQSZA * WEEKS) + (EQSZB * WEEKS*WEEKS)

      ELSE
         STOP  'ERROR WITH ORGANISM TYPE IN FUNCTION AGESZE'
      ENDIF


      RETURN
      END






      BLOCK DATA BLK1
C     ---------------

C  ORGNSM:
C  -------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES MUST BE DATA INITIALIZED:
C    NOOFT(N)
C    OPQOK
C    OPNOK
C    OPN$OK(N)
C    OPTOK(N)
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE: INITIALIZE VARIABLES IN COMMON BLOCKS, ICBLK1

C  1. PARAMETERS
      INCLUDE  (IPAR1)
      INCLUDE  (ICUNIT)
C  3. COMMON BLOCKS
      INCLUDE (ICBLK1)
C  4. LOCAL VARIABLES
      INTEGER  I, J, K, M, N, T

C  NAME   DESCRIPTION OF LOCAL VARIABLES
C  ------ -------------------------------------------------------------
C  I      LOOP COUNTER FOR STAGES
C  J      LOOP COUNTER FOR PERIODS
```

```
C K       LOOP COUNTER FOR PROGRAM RUNNING OPTIONS
C M       LOOP COUNTER FOR MONTHS
C N       LOOP COUNTER FOR INPUTS
C T       LOOP COUNTER FOR TYPES OF INPUT N
C --------------------------------------------------------------------


C VARIABLES THAT SHOULD NOT CHANGE
C --------------------------------
C  NAME OF MONTHS
      DATA (NAMEM(M), M=1,12) /'JANUARY','FEBRUARY','MARCH','APRIL',
     2                          'MAY','JUNE','JULY','AUGUST','SEPTEMBER',
     3                          'OCTOBER','NOVEMBER','DECEMBER' /

C  UNIT NUMBERS FOR ALL BUT OUTPUT FILES/ AND TO INITIATE VARIABLES
C  ----------------------------------------------------
      DATA  WF, RI, WI, RF / 3,5,6,8 /
      DATA (QOUTQN(N), N=1,NOOFN) / NOOFN * 'N' /


C VARIABLES THAT CHANGE WITH CAPABILITIES OF THE PROGRAM'S OPTIONS
C ---------------------------------------------------------------
C  NUMBER OF TYPES,T, OF EACH INPUT,N
      DATA (NOOFT(N), N=1,NOOFN) / 1,1,1,1,1,1 /

C  OPTION INFORMATION:  0=AVAILABLE, 1=NOT AVAILABLE
C     OPTIONS 1-3 FOR RUNNING PROGRAM
      DATA (OPQOK(K), K=1,3) / 0,0,1 /

C     TO COMPUTE QUANTITY FOR EACH INPUT,N, IN EACH STAGE,J
      DATA (OPNOK(I,1), I=1,NOOFI) / NOOFI * 1 /
      DATA (OPNOK(I,2), I=1,NOOFI) / NOOFI * 1 /
      DATA (OPNOK(I,3), I=1,NOOFI) / NOOFI * 1 /
      DATA (OPNOK(I,4), I=1,NOOFI) / NOOFI * 1 /
      DATA (OPNOK(I,5), I=1,NOOFI) / 1,0,0,0,0 /
      DATA (OPNOK(I,6), I=1,NOOFI) / NOOFI * 1 /

C     TO COMPUTE THE PRICE OF INPUT,N
      DATA (OPN$OK(N), N=1,NOOFN) / 1,1,1,1,0,1 /

C     TO CHANGE THE NUMBER OF TYPES,T, OF EACH INPUT,N
      DATA (OPTOK(N), N=1,NOOFN) / 1,1,0,1,0,1 /

C     TO PRINT OUT ADDITIONAL INFORMATION ABOUT PRODUCTION INPUTS
       DATA (OUTQN(N), N=1,NOOFN) / 1,1,1,1,0,1 /



      END




      BLOCK DATA BLKO
C     ---------------

C  ORGNSM:
C  ------
C  SUBROUTINE VARIES WITH ORGANISM
C  VARIABLES THAT MUST BE DATA INITIALIZED:
C    NAMEN(N)
C    NAMENU(N)
C    NAMEI(I)
C    NAMET(N,T)
C    NAMEO(O)
C  CURRENT ORGANISM:  BIVALVES

C  STRUCTURE:  INITIALIZE VARIABLES IN COMMON BLOCK, ICBLKO

C  1. PARAMETERS
      INCLUDE (IPAR1)
C  3. COMMON BLOCKS
      INCLUDE (ICBLKO)
C  4. LOCAL VARIABLES
      INTEGER  EQ, I, N, O, T

C  NAME    DESCRIPTION OF LOCAL VARIABLES
C  ------  ------------------------------------------------------------
C  EQ      LOOP COUNTER FOR FEEDING EQUATIONS, INPUT 5
C  I       LOOP COUNTER FOR STAGES
C  N       LOOP COUNTER FOR INPUTS
C  O       LOOP COUNTER FOR ORGANISM TYPES
C  T       LOOP COUNTER FOR TYPES OF INPUT N
C  --------------------------------------------------------------------


C  VARIABLES REQUIRED BUT THAT CHANGE WITH ORGANISM:
C  ------------------------------------------------
      DATA  NAME / 'BIVALVE' /

      DATA (NAMEN(N), N=1,NOOFN) /'ELECTRICTY','FUEL OIL','LABOR',
     2                            'MATERIALS','ALGAE','CULCH' /

      DATA (NAMENU(N), N=1,NOOFN) /'KWHR','LITRE','HOUR','UNIT',
     2                            'CELL','LITRE' /

      DATA (NAMEI(I), I=1,NOOFI) / 'CONDITION', 'SPAWN','LARVAE',
     2                             'SETTING', 'JUVENILE' /

      DATA (NAMEO(O), O=1,NOOFO) / 'OYSTERS', 'CLAMS' /


C VARIABLES RELATING TO ORGANISM DEPENDENT FUNCTIONS:
C ---------------------------------------------------
      DATA EQNO(1)/'POPULATION = POPULATION(INITIAL) * [(WEEKS+1)**A]'/

      DATA EQNOAF / -.255068 /

      DATA (EQQ5(EQ),EQ=1,2)/
     2     'NATURAL LOG(SIZE IN MM)  =  A + B * NAT LOG(10,000 CELLS)',
     2     'NATURAL LOG(AGE IN WKS)  =  A + B * NAT LOG(10,000 CELLS)'/

      DATA (EQQ5AF(EQ),EQ=1,2) / -.631048, -.330940 /
      DATA (EQQ5BF(EQ),EQ=1,2) / .338426, .344505 /

      DATA EQSZE(1)  / 'SIZE = (A * AGE) + (B * AGE SQUARED)' /

      DATA EQSZAF / .737 /
      DATA EQSZBF / -.00252 /



      END
```

```
C  FILE NAME:  ICN5$
C  ORGNSM:  VARIABLES SPECIFIC TO ORGANISM
C  CURRENT ORGANISM:  BIVALVES
C  TO INCLUDE COMMONS USED IN PRICE5 SUBROUTINE
C  VARIABLE NAMES ARE IDENTICAL IN ALL PRICE5 SUBROUTINES.
C  PARAMETERS USED IN PRICE5
      INTEGER  NUMGOD, MAXLAB
      PARAMETER (NUMGOD=2, MAXLAB=6)
      COMMON / CN5$ /
     I BEGPRO, ENDPRO, NUMLAB, NUMSTG,
     R BLBLIF, COMEFF, HPCOM, HPPMP, HRSCOM, NUMTUB,
     R PKWHR, PMPEFF, PMPGPM, POIL, WATAIR, WATAUT, WATBLB,
     R ALLAB(MAXT),BLBRM(MAXT),CALIN(MAXT),TCELL(MAXT),CPERML(MAXT),
     R DAYSTG(MAXT),DISSTG(MAXT),HRSAUT(MAXT),HRSBLB(MAXT),H2OTNK(MAXT),
     R INQTNK(MAXT),LFOOD(MAXT),LITWHR(MAXT),NUMAIR(MAXT),NUMTNK(MAXT),
     R STGRM(MAXT),TALLAB(MAXT),TALLAL(MAXT),TCALIN(MAXT),
     R TKWHR(MAXT),TLITRE(MAXT),TOIL(MAXT),TTKWHL(MAXT),TTKWHR(MAXT),
     R TTOIL(MAXT),TTOILL(MAXT),NUTLTR(MAXT),TUBTNK(MAXT),
     R ALFOOU(MAXT),TCCELL(MAXT),
     R HRLABW(6),PLABOR(6),TCLATY(6),TLABTY(6,MAXT),TTLATY(6,MAXT),
     R HRLABS(6,MAXT),PGOOD(5),TGOTY(5,MAXT),TTGOTY(5,MAXT),
     R CINP(4,MAXT),TCINP(4,MAXT),TCINPL(4,MAXT),
     R HRSAIR(MAXM,MAXT),TCLITR(MAXT),
     C NAMALG(MAXT),MONAVA(MAXT),
     C DATE2, QWRT$5
      INTEGER BEGPRO, ENDPRO, NUMLAB, NUMSTG
      REAL BLBLIF, COMEFF, HPCOM, HPPMP, HRSCOM, NUMTUB, PKWHR,
     R      PMPEFF, PMPGPM, POIL, WATAIR, WATAUT, WATBLB,
     R      ALLAB, BLBRM, CALIN, TCELL, CPERML, DAYSTG, DISSTG, HRSAUT,
     R      HRSBLB, H2OTNK, INQTNK, LFOOD, LITWHR, NUMAIR, NUMTNK,
     R      STGRM, TALLAB, TALLAL, TCALIN, TKWHR, TLITRE, TOIL,
     R      TTKWHL, TTKWHR, TTOIL, TTOILL, NUTLTR, TUBTNK,
     R      ALFOOU, TCCELL,
     R      HRLABW, PLABOR, TCLATY, TLABTY, TTLATY,
     R      PGOOD, TGOTY, TTGOTY, CINP, TCINP, TCINPL, HRLABS,
     R      HRSAIR, TCLITR
      CHARACTER*10 NAMALG, MONAVA
      CHARACTER*40 DATE2
      CHARACTER*1  QWRT$5
```

```
C  FILE NAME:  ICBLKO
C  ORGNSM:  VARIABLE DON'T CHANGE
C  FOR INCLUSION OF COMMON BLOCK DATA INITIALIZED, ORGANISM SPECIFIC
      COMMON / CBLKO /
     R  EQNOA, EQNOAF, EQQ5A, EQQ5AF, EQQ5B, EQQ5BF,
     R  EQSZA, EQSZAF, EQSZB, EQSZBF,
     C  EQNO, EQQ5, EQSZE,
     C  NAME, NAMEO, NAMEI, NAMEN, NAMET, NAMENU
      REAL  EQNOA, EQNOAF, EQQ5A, EQQ5AF, EQQ5B, EQQ5BF,
     R     EQSZA, EQSZAF, EQSZB, EQSZBF
      CHARACTER *  5  NAMENU
      CHARACTER * 10  NAME, NAMEO, NAMEI, NAMEN, NAMET
      CHARACTER * 62  EQNO, EQQ5, EQSZE
      DIMENSION
     R  EQQ5A(2), EQQ5AF(2), EQQ5B(2), EQQ5BF(2),
     C  EQNO(1), EQQ5(2), EQSZE(1),
     C  NAMEO(NOOFO), NAMEI(NOOFI), NAMEN(NOOFN),
     C  NAMET(NOOFN,MAXT), NAMENU(NOOFN)
```

```
C  FILE NAME : ICMAIN
C  ORGNSM:  VARIABLES DON'T CHANGE
C  TO INCLUDE COMMONS NECESSARY TO RUN BIVALVE PROGRAM SKELETON
      COMMON / CMAIN/
     I   ANYNOK, BEGINI, BEGINM, BEGINJ, OTYPE, ENDJ,
     I   ENDM, IOFJ, JPERI, MOFJ, M1, M2, OPTION, PASEDM, TOTALI,
     R   AGEUPJ, ONOBEG, ONOOFJ, ONOUPJ, CSOOFJ, CSOUPJ, CSNOFJ,
     R   CSNUPJ, CSOFJ, CSOP1, CSUPJ,
     R   DAYUPJ, DAYOP1, DAYSI, DAYSJ, PRICET, SZEOFJ, SZEUPJ,
     R   TQUANI, TQUANJ, WKSUPJ,
     C   DATE, NAMEF1, NAMES, NQBYM, N$BYM, OPNNEW, QOPN$, QOUTQN
      INTEGER
     I   ANYNOK, BEGINI, BEGINM, BEGINJ, OTYPE, ENDJ,
     I   ENDM, IOFJ, JPERI, MOFJ, M1, M2, OPTION, PASEDM, TOTALI
      REAL
     R   AGEUPJ, ONOBEG, ONOOFJ, ONOUPJ, CSOOFJ, CSOUPJ, CSNOFJ,
     R   CSNUPJ, CSOFJ, CSOP1, CSUPJ, DAYUPJ,
     R   DAYOP1, DAYSI, DAYSJ, PRICET, SZEOFJ, SZEUPJ,
     R   TQUANI, TQUANJ, WKSUPJ
      CHARACTER * 50  DATE, NAMES
      CHARACTER * 10  NAMEF1
      CHARACTER * 1   NQBYM, N$BYM, QOPN$, OPNNEW, QOUTQN
      DIMENSION
     I   ANYNOK(NOOFN), BEGINJ(NOOFI), IOFJ(MAXJI), JPERI(NOOFI),
     I   MOFJ(MAXJI), PASEDM(MAXJI),
     R   AGEUPJ(MAXJI), ONOOFJ(MAXJI), ONOUPJ(MAXJI), CSOOFJ(MAXJI),
     R   CSOUPJ(MAXJI), CSNOFJ(MAXJI,NOOFN), CSNUPJ(MAXJI,NOOFN),
     R   CSOFJ(MAXJI),
     R   CSUPJ(MAXJI), DAYUPJ(MAXJI), DAYSI(NOOFI),
     R   DAYSJ(NOOFI), ENDJ(NOOFI), PRICET(NOOFN,MAXT,MAXM),
     R   SZEOFJ(MAXJI), SZEUPJ(MAXJI),
     R   TQUANI(NOOFI,NOOFN,MAXT,MAXM), TQUANJ(MAXJI,NOOFN,MAXT),
     C   NQBYM(NOOFN), N$BYM(NOOFN), OPNNEW(NOOFI,NOOFN), QOPN$(NOOFN),
     C   QOUTQN(NOOFN)
```

```
C  FILE NAME : ICBLK1
C  ORGNSM:  VARIABLES DON'T CHANGE
C  USED FOR INCLUSION OF COMMON BLOCKS VARIABLES INITIALIZED IN BLOCK
C  DATA SUBPROGRAM
      COMMON / CBLK1 /
     I   NOOFT, OPTOK, OPN$OK, OPQOK, OPNOK, OUTQN,
     C   NAMEM
      INTEGER   NOOFT, OPTOK, OPN$OK, OPQOK, OPNOK, OUTQN
      CHARACTER * 10  NAMEM
      DIMENSION
     I   NOOFT(NOOFN), OPTOK(NOOFN), OPN$OK(NOOFN), OPQOK(3),
     I   OPNOK(NOOFI,NOOFN), OUTQN(NOOFN),
     C   NAMEM(MAXM)
```

```
C  FILE  NAME :  ICUNIT
C  ORGNSM: VARIABLES DON'T CHANGE WITH ORGANISM, BUT WITH MACHINE CAPABI
C  ALL UNIT NUMBER FOR READ, WRITE, AND OUTPUT FILES
      COMMON / CUNIT /
     I   RF, RI, WF, WI, WO, WON
      INTEGER  RF, RI, WF, WI, WO, WON
      DIMENSION WON(NOOFN)
```

```
C  FILE NAME:  ICORG
C  ORGNSM: VARIABLES SPECIFIC TO ORGANISM
C  CURRENT ORGANISM:  BIVALVE
C  TO INCLUDE COMMONS SPECIFIC TO ORGANISM
      COMMON / CORG /
     I   QEQQ5,
     R   DAYSI1, FILTER, NOOI1, N5AVAL, N5REQ, QUAOFJ, SEACEL,
     C   N5ABYM, SEABYM
      INTEGER  QEQQ5
      REAL  DAYSI1, FILTER, NOOI1, N5AVAL, N5REQ, QUAOFJ, SEACEL
      CHARACTER*1  N5ABYM, SEABYM
      DIMENSION
     R   FILTER(MAXI), N5AVAL(MAXT,MAXM),
     R   N5REQ(MAXI,MAXT), QUAOFJ(MAXJI,MAXT), SEACEL(MAXI,MAXM)
```

```
C  FILE NAME : IPAR1
C  ORGNSM: VARIABLES DON'T CHANGE; PARAMETERS ARE SPECIFIC TO ORGANISM
C  CURRENT ORGANISM:  BIVALVE
C  FOR INCLUSION OF PARAMETERS
      INTEGER
     I   MAXI, MAXJ, MAXJI, MAXM, MAXN, MAXT,
     I   NOOFO, NOOFI, NOOFN
      PARAMETER
     I   (MAXI = 6, MAXJ = 10, MAXM = 12, MAXN = 6, MAXT = 6,
     I    MAXJI = MAXI * MAXJ,
     I    NOOFO = 2, NOOFI = 5, NOOFN = 6)
```

APPENDIX B

This appendix contains the data used in the base run of the simulation model and the output for the base run. All or portions of this information is printed at the user's request. However, what appears in the following pages is an edited version of what is actually printed by the program. Data were not available to estimate all component costs of bivalve seed. Thus, to save space, only that information pertaining to algae and total feed costs is reported. In addition, the algae price computations are reported only for one month, although algae cost differences by month are reflected in the bivalve feed costs reported in subsequent sections of the output.

```
BIVALVE    COST SIMULATION MODEL:  INPUTS ENTERED
 -------------------------------------------------
 DATE:  MARCH 30, 1988
 OPTION DATA ENTERED
 NAME OF FILE USED:  1201 8DATA
 OPTION CHOSEN: 1
               COST UP TO FIRST STAGE OF INTEREST IS ENTERED
               USER CHOOSES INPUTS TO COMPUTE IN REMAINING STAGES
               THE FIRST STAGE OF INTEREST IS THE SPAWN     STAGE
 IF C, THE PROGRAM COMPUTES THE QUANTITY
 IF E, THE USERS ENTERS THE QUANTITY
 IF -, THE QUANTITY IS NEITHER ENTERED NOR COMPUTED
 INPUT        CONDITION  SPAWN      LARVAE     SETTING    JUVENILE
 ELECTRICTY   -          E          E          E          E
 FUEL OIL     -          E          E          E          E
 LABOR        -          E          E          E          E
 MATERIALS    -          E          E          E          E
 ALGAE        -          C          C          C          C
 CULCH        -          E          E          E          E
 THE NUMBER OF EACH SECTION FOR WHICH DATA MUST BE ENTERED.
   SECTION  1: GENERAL    DATA
   SECTION  2: BIVALVE    DATA
   SECTION  7: ALGAE      DATA
BIVALVE    COST SIMULATION MODEL:  ALGAE PRICE COMPUTATIONS
 -----------------------------------------------------------
 DATE:  MARCH 30, 1988   INPUTS ENTERED
 1.   ALGAE THESIS RESULTS
*PRICE INFORMATION
2. OF ELECTRICITY                    $    0.1000   PER KWHR
3. OF FUEL OIL                       $    0.2990   PER LITR
4. OF LIGHTBULBS                     $    2.0000   PER BULB
4. OF NUTRIENT SOL.                  $    1.0000   PER LITR
5. # OF LABOR TYPES:  3
6. PRICE OF TYPE1                    $    6.7310   PER HR
6. PRICE OF TYPE2                    $    5.5290   PER HR
6. PRICE OF TYPE3                    $    4.8080   PER HR
*MISC. INFORMATION - NON STAGE DEPENDENT
8.  WATT RATING OF AIR CONDIT.            1050.0      WATT
9.  WATT RATING OF LITEBULBS                40.0      WATT
10. LIFETIME OF LITEBULBS              15000.0      HRS
11. WATT RATING OF AUTOCLAVE            2000.0      WATT
12. COMPRESSOR HORSEPOWER                  1.0      HP
13. HOURS COMPRESSOR ON/DAY               24.0      HRS
14. TOTAL # COMPRESSOR TUBES               74.      TUBE
15. EFF. RATING OF COMPRESSOR           0.850
16. EFF. RATING OF WATER PUMP           0.450
17. GPM RATING OF WATER PUMP             40.0       GPM
18. HP RATING OF WATER PUMP               2.5       HP
19. HOURS OF LABOR/WEEK
    OF TYPE1                              11.5       HRS
    OF TYPE2                               3.0       HRS
    OF TYPE3                               0.0       HRS
```

*STAGE INFORMATION
20. NUMBER OF STAGES: 6

	1STAGE	2STAGE	3STAGE	4STAGE	5STAGE	6STAGE
21. NUTRIENT/LITRE H2O	0.100E-02	0.225E-02	0.101E-02	0.269E-03	0.147E-03	0.587E-04
22. NUMBER OF TANKS	6.0	5.0	5.0	5.0	5.0	2.0
23. NUMBER OF STAGES PER ROOM	3.0	3.0	3.0	1.0	1.0	1.0
24. DAYS TO REACH MATURITY	14.0	7.0	7.0	7.0	7.0	7.0
25. CELLS PER MILLITRE - DENSITY	0.300E+06	0.300E+06	0.300E+06	0.400E+06	0.650E+06	0.600E+06
26. LITRES WATER ADDED PER TANK	0.500E-02	1.24	13.8	743.	0.341E+04	0.341E+05
27. LITRES OF INNOCULATION ADDED/TANK	0.000E+00	0.500E-02	0.200	14.0	379.	0.379E+04
28. AVG. LITRES UNCONTROLLED DISCARD	0.500E-02	2.08	0.000E+00	379.	0.170E+04	0.852E+04
29. NUMBER OF AIR TUBES PER TANK	1.00	1.00	1.00	2.00	4.00	5.00
30. NUMBER OF LIGHTBULBS PER ROOM	10.0	10.0	10.0	12.0	8.0	8.0
31. HOURS LIGHTS ON PER DAY PER ROOM	24.00	24.00	24.00	24.00	2.00	0.50
32. HOURS AUTOCLAVE USED	0.17	0.50	0.84	0.00	0.00	0.00
33. NUMBER OF AIR CONDIT. PER ROOM	1.0	1.0	1.0	1.0	0.0	0.0
34. HRS OF LABOR PER STAGE						
OF TYPE1	1.50	1.50	1.50	4.00	2.00	2.00
OF TYPE2	0.00	0.00	0.00	0.00	2.00	2.00
OF TYPE3	0.00	0.00	0.00	0.00	0.00	2.00
35. NAME OF EACH ALGAE STAGE	PRIMARY1	PRIMARY2	PRIMARY3	SECONDARY	TERTIARY	SOLAR

*MONTH INFORMATION
36. MONTH BEGIN PRODUCTION 1 JANUARY
37. MONTH END PRODUCTION 12 DECEMBER

	1STAGE	2STAGE	3STAGE	4STAGE	5STAGE	6STAGE
38. HOURS AIR CONDIT. ON PER DAY/ROOM						
1 JANUARY	0.00	0.00	0.00	0.00	0.00	0.00
2 FEBRUARY	0.00	0.00	0.00	0.00	0.00	0.00
3 MARCH	0.00	0.00	0.00	0.00	0.00	0.00
4 APRIL	4.40	4.40	4.40	4.40	0.00	0.00
5 MAY	8.80	8.80	8.80	8.80	0.00	0.00
6 JUNE	11.50	11.50	11.50	11.50	0.00	0.00
7 JULY	16.00	16.00	16.00	16.00	0.00	0.00
8 AUGUST	15.00	15.00	15.00	15.00	0.00	0.00
9 SEPTEMBER	11.50	11.50	11.50	11.50	0.00	0.00
10 OCTOBER	8.40	8.40	8.40	8.40	0.00	0.00
11 NOVEMBER	4.40	4.40	4.40	4.40	0.00	0.00
12 DECEMBER	0.00	0.00	0.00	0.00	0.00	0.00

BIVALVE COST SIMULATION MODEL: ALGAE PRICE COMPUTATIONS (cont.)

DATE: MARCH 30, 1988
OUTPUT GENERATED
DATE OF PRICE COMPUTATIONS: ALGAE THESIS RESULTS

BATCH BEGUN: APRIL

	1STAGE	2STAGE	3STAGE	4STAGE	5STAGE	6STAGE
ALGAE AVAILABLE FOR USE IN MONTH	APRIL	APRIL	APRIL	MAY	MAY	MAY
*TOTAL LITRES PRODUCED	0.2500E-01	4.170	70.00	3406.	0.1722E+05	0.6718E+05
LITRES OF ALGAE AVAILABLE	0.0000E+00	3.170	0.0000E+00	1511.	9652.	0.6718E+05
CELLS OF ALGAE PRODUCED	0.7500E+07	0.1251E+10	0.2100E+11	0.1362E+13	0.1119E+14	0.4031E+14
*TOTAL COST OF STAGE	50.64	80.90	49.79	110.3	109.3	107.1
TOTAL COST PER LITRE OF ALGAE	2025.	19.40	0.7113	0.3240E-01	0.6347E-02	0.1594E-02
TOTAL COST PER CELL OF ALGAE	0.6752E-05	0.6467E-07	0.2371E-08	0.8099E-10	0.9764E-11	0.2656E-11
TOTAL COST OF STAGE, LESS INNOCULATION	50.64	30.26	30.39	60.55	47.91	59.01
*TOTAL COST OF ELECTRICITY	9.060	13.47	7.716	24.41	18.79	13.47
TOTAL COST OF ELECTRICITY/LITRE OF ALG	362.4	3.231	0.1102	0.7166E-02	0.1091E-02	0.2005E-03
TOTAL COST OF ELEC. LESS INNNOCULATION	9.060	4.414	4.484	16.69	5.208	5.210
*TOTAL KWHR USED	90.60	134.7	77.16	244.1	187.9	134.7
TOTAL KWHR USED PER LITRE OF ALGAE	3624.	32.31	1.102	0.7166E-01	0.1091E-01	0.2005E-02
TOTAL KWHR USED LESS INNOCULATION	90.60	44.14	44.84	166.9	52.08	52.10
*TOTAL COST OF FUEL OIL	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
TOTAL COST OF OIL/LITRE OF ALGAE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
TOTAL COST OF OIL LESS INNOCULATION	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
*TOTAL LITRES OF FUEL OIL	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
TOTAL LITRES OIL USED/LITRE OF ALGAE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
TOTAL L. OIL USED LESS INNOCULATION	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
*TOTAL COST OF ALL LABOR TYPES	41.43	67.19	41.87	84.46	87.18	88.12
TOTAL COST ALL LABOR/LITRE OF ALGAE	1657.	16.11	0.5982	0.2480E-01	0.5062E-02	0.1312E-02
TOTAL COST ALL LABOR LESS INNOCULATN	41.43	25.76	25.76	42.59	40.19	49.80
*TOTAL HOURS OF ALL LABOR TYPES	6.333	10.25	6.375	12.79	13.53	14.37
TOTAL HRS ALL LABOR USED/LITRE ALGAE	253.3	2.458	0.9107E-01	0.3756E-02	0.7858E-03	0.2138E-03
TTL HRS ALL LABOR LESS INNOCULATION	6.333	3.917	3.917	6.417	6.417	8.417
*TOTAL COST OF ALL MATERIALS	0.1494	0.2380	0.2017	1.470	3.333	5.468
TTL COST ALL MATERIALS/LITRE OF ALGAE	5.975	0.5708E-01	0.2882E-02	0.4315E-03	0.1935E-03	0.8139E-04
TTL COST ALL MATER. LESS INNOCULATION	0.1494	0.8867E-01	0.1446	1.268	2.515	4.003

BIVALVE COST SIMULATION MODEL: INPUT ENTERED (CONT.) MARCH 30, 1988

GENERAL DATA

1. MONTH THAT PRODUCTION SEASON BEGINS: JANUARY (1)
2. MONTH THAT PRODUCTION SEASON ENDS: DECEMBER (12)

2 STAGE	3 STAGE	4 STAGE	5 STAGE

3. NUMBER OF DAYS IN EACH STAGE:

2.00	12.00	5.00	371.00

4. NUMBER OF PERIODS IN EACH STAGE:

1	3	1	53

5. FOR OPTION 1, THE NUMBER OF DAYS UP TO STAGE 2: 30.00
6. FOR OPTION 1, THE COST UP TO STAGE 2: 0.00

INFORMATION PERTAINING TO PRICE

INPUT: ELECTRICTY
7. PRICE COMPUTED BY PROGRAM?: N

INPUT: FUEL OIL
8. PRICE COMPUTED BY PROGRAM?: N

INPUT: LABOR
9. PRICE COMPUTED BY PROGRAM?: N

INPUT: MATERIALS
10. PRICE COMPUTED BY PROGRAM?: N

INPUT: ALGAE
11. PRICE COMPUTED BY PROGRAM?: Y

INPUT: CULCH
12. PRICE COMPUTED BY PROGRAM?: N

INPUT: ALGAE
32. PRICE OF ALGAE PER CELL (COMPUTED BY THE PROGRAM):

	JANUARY JULY	FEBRUARY AUGUST	MARCH SEPTEMBER	APRIL OCTOBER	MAY NOVEMBER	JUNE DECEMBER
PRIMARY1 :						
	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6751600E-05	0.7039068E-05	0.7215468E-05
	0.7509468E-05	0.7444134E-05	0.7215468E-05	0.7012933E-05	0.6751600E-05	0.6464135E-05
PRIMARY2 :						
	0.6208467E-07	0.6208467E-07	0.6208467E-07	0.6466979E-07	0.6725492E-07	0.6884125E-07
	0.7148515E-07	0.7089761E-07	0.6884125E-07	0.6701993E-07	0.6466979E-07	0.6208467E-07
PRIMARY3 :						
	0.2282785E-08	0.2282785E-08	0.2282785E-08	0.2371048E-08	0.2459312E-08	0.2513474E-08
	0.2603744E-08	0.2583683E-08	0.2513474E-08	0.2451288E-08	0.2371048E-08	0.2282785E-08
SECONDARY :						
	0.7488353E-10	0.7488353E-10	0.7488353E-10	0.7725728E-10	0.8099153E-10	0.8380865E-10
	0.8707120E-10	0.8792311E-10	0.8572570E-10	0.8297109E-10	0.7985458E-10	0.7624403E-10
TERTIARY :						
	0.9350308E-11	0.9350308E-11	0.9350308E-11	0.9511042E-11	0.9763898E-11	0.9954654E-11
	0.1017557E-10	0.1023326E-10	0.1008446E-10	0.9897940E-11	0.9686914E-11	0.9442431E-11
SOLAR :						
	0.2605360E-11	0.2605360E-11	0.2605360E-11	0.2624981E-11	0.2655846E-11	0.2679130E-11
	0.2706097E-11	0.2713139E-11	0.2694976E-11	0.2672208E-11	0.2646449E-11	0.2616606E-11

```
BIVALVE    COST SIMULATION MODEL:  INPUT ENTERED (CONT.)      MARCH 30, 1988
 BIVALVE    DATA

  1. TYPE OF BIVALVE                            1 =OYSTERS
  2. NAME OF SYSTEM                            UPWELL
  3. NUMBER DAYS IN STAGE CONDITION            30.000
  4. NUMBER BIVALVES IN STAGE CONDITION        2.0000
  5. NUMBER OF OYSTERS    BATCH BEGUN WITH:   1000000.

 EQUATION TO DETERMINE SURVIVAL:
     POPULATION = POPULATION(INITIAL) * [(WEEKS+1)**A]
  6. PARAMETER A = -0.2550680

 EQUATION TO EQUATE AGE TO SIZE:
     SIZE = (A * AGE) + (B * AGE SQUARED)
  7. PARAMETER A =  0.7370000
  8. PARAMETER B = -0.2520000E-02

BIVALVE    COST SIMULATION MODEL:  INPUT ENTERED (CONT.)      MARCH 30, 1988
 ALGAE      DATA

  1. AGE/ALGAE EQUATION CHOSEN: 1
     NATURAL LOG(SIZE IN MM)  =  A + B * NAT LOG(10,000 CELLS)
  2. PARAMETER A = -0.6310480
  3. PARAMETER B =  0.3384260

 IN THE FOLLOWING, WHERE QUANTITY=0, QUANTITY IS ENTERED

                                                2 STAGE   3 STAGE   4 STAGE   5 STAGE
  4. FILTERING RATE                              0.50      0.50      0.50      0.50

  5. FRACTION OF EACH ALGAE TYPE REQUIRED  PER STAGE:
      TYPE PRIMARY1                              0.00E+00  0.00E+00  0.00E+00  0.00E+00
      TYPE PRIMARY2                              0.00E+00  0.00E+00  0.00E+00  0.00E+00
      TYPE PRIMARY3                              0.00E+00  0.00E+00  0.00E+00  0.00E+00
      TYPE SECONDARY                             0.00E+00  0.00E+00  0.00E+00  0.00E+00
      TYPE TERTIARY                              1.0       1.0       1.0      0.00E+00
      TYPE SOLAR                                 0.00E+00  0.00E+00  0.00E+00   1.0

  6. FRACTION FROM SEA VARIES BY MONTH?: N
  7. FRACTION OF DIET FROM SEA ALGAE IN EACH STAGE, EACH MONTH:
       JANUARY                                   0.00E+00  0.00E+00  0.00E+00  0.12
       FEBRUARY                                  0.00E+00  0.00E+00  0.00E+00  0.12
       MARCH                                     0.00E+00  0.00E+00  0.00E+00  0.12
       APRIL                                     0.00E+00  0.00E+00  0.00E+00  0.12
       MAY                                       0.00E+00  0.00E+00  0.00E+00  0.12
       JUNE                                      0.00E+00  0.00E+00  0.00E+00  0.12
       JULY                                      0.00E+00  0.00E+00  0.00E+00  0.12
       AUGUST                                    0.00E+00  0.00E+00  0.00E+00  0.12
       SEPTEMBER                                 0.00E+00  0.00E+00  0.00E+00  0.12
       OCTOBER                                   0.00E+00  0.00E+00  0.00E+00  0.12
       NOVEMBER                                  0.00E+00  0.00E+00  0.00E+00  0.12
       DECEMBER                                  0.00E+00  0.00E+00  0.00E+00  0.12
```

BIVALVE COST SIMULATION MODEL: OUTPUT GENERATED

DATE: MARCH 30, 1988
FILE: 1201 8DATA

BIVALVE TYPE: OYSTERS
NUMBER BIVALVE BEGUN: .10000E+07
SYSTEM USED: UPWELL
OPTION CHOSEN: 1, FIRST STAGE OF INTEREST IS THE SPAWN STAGE
(*NOTE: STAGE 1 INCLUDES ALL STAGES UP TO THAT STAGE)

PRICE OF INPUTS PER MONTH

MONTH	JANUARY JULY	FEBRUARY AUGUST	MARCH SEPTEMBER	APRIL OCTOBER	MAY NOVEMBER	JUNE DECEMBER
ELECTRICTY/KWHR	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
FUEL OIL /LITRE	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
LABOR /HOUR	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
MATERIALS /UNIT	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
PRIMARY1 /CELL	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6751600E-05	0.7039068E-05	0.7215468E-05
	0.7509468E-05	0.7444134E-05	0.7215468E-05	0.7012933E-05	0.6751600E-05	0.6464135E-05
PRIMARY2 /CELL	0.6208467E-07	0.6208467E-07	0.6208467E-07	0.6466979E-07	0.6725492E-07	0.6884125E-07
	0.7148515E-07	0.7089761E-07	0.6884125E-07	0.6701993E-07	0.6466979E-07	0.6208467E-07
PRIMARY3 /CELL	0.2282785E-08	0.2282785E-08	0.2282785E-08	0.2371048E-08	0.2459312E-08	0.2513474E-08
	0.2603744E-08	0.2583683E-08	0.2513474E-08	0.2451288E-08	0.2371048E-08	0.2282785E-08
SECONDARY /CELL	0.7488353E-10	0.7488353E-10	0.7488353E-10	0.7725728E-10	0.8099153E-10	0.8380865E-10
	0.8707120E-10	0.8792311E-10	0.8572570E-10	0.8297109E-10	0.7985458E-10	0.7624403E-10
TERTIARY /CELL	0.9350308E-11	0.9350308E-11	0.9350308E-11	0.9511042E-11	0.9763898E-11	0.9954654E-11
	0.1017557E-10	0.1023326E-10	0.1008446E-10	0.9897940E-11	0.9686914E-11	0.9442431E-11
SOLAR /CELL	0.2605360E-11	0.2605360E-11	0.2605360E-11	0.2624981E-11	0.2655846E-11	0.2679130E-11
	0.2706097E-11	0.2713139E-11	0.2694976E-11	0.2672208E-11	0.2646449E-11	0.2616606E-11
CULCH /LITRE	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

BIVALVE COST SIMULATION MODEL: OUTPUT GENERATED (CONT.) MARCH 30, 1988 BATCH BEGUN: APRIL

PRD	*STG	NAME OF STAGE	MNTH	TIME UP TO END OF PERIOD	AGE AT END OF PERIOD	AVG NO. BIVALV IN PERIOD	NO. BIVALV AT END OF PERIOD	AVG SIZE IN PERIOD	SIZE BIVALV AT END PERIOD
1	1	CONDITION	4	30.00000	730.0000	2.000000	2.000000	49.45227	49.45227
2	2	SPAWN	5	32.00000	2.000000	967734.0	937909.1	0.1052085	0.2103657
3	3	LARVAE	5	36.00000	6.000000	892905.5	853937.6	0.3939469	0.6298628
4	3	LARVAE	5	40.00000	10.00000	824216.0	797460.8	0.7601836	1.047713
5	3	LARVAE	5	44.00000	14.00000	775672.3	755616.8	1.249980	1.463920
6	4	SETTING	5	49.00000	19.00000	734727.4	715555.1	1.676703	1.981861
7	5	JUVENILE	5	56.00000	26.00000	693416.0	673338.0	2.272539	2.702662
8	5	JUVENILE	6	63.00000	33.00000	656581.6	641096.2	3.010759	3.418421
9	5	JUVENILE	6	70.00000	40.00000	627751.0	615260.1	3.744302	4.129142
10	5	JUVENILE	6	77.00000	47.00000	604251.8	593853.4	4.473164	4.834821
11	5	JUVENILE	6	84.00000	54.00000	584536.4	575674.4	5.125919	5.535461
12	5	JUVENILE	7	91.00000	61.00000	567632.6	559942.1	5.845417	6.231061
13	5	JUVENILE	7	98.00000	68.00000	552892.6	546121.6	6.560244	6.921621
14	5	JUVENILE	7	105.0000	75.00000	539864.1	533832.3	7.198956	7.607141
15	5	JUVENILE	7	112.0000	82.00000	528219.8	522793.8	7.904419	8.287622
16	5	JUVENILE	7	119.0000	89.00000	517716.3	512794.6	8.605203	8.963061
17	5	JUVENILE	8	126.0000	96.00000	508166.7	503671.1	9.229876	9.633462
18	5	JUVENILE	8	133.0000	103.0000	499426.1	495294.5	9.921297	10.29882
19	5	JUVENILE	8	140.0000	110.0000	491378.7	487561.6	10.60804	10.95914
20	5	JUVENILE	8	147.0000	117.0000	483931.8	480388.5	11.21867	11.61442
21	5	JUVENILE	9	154.0000	124.0000	477009.6	473706.4	11.89605	12.26466
22	5	JUVENILE	9	161.0000	131.0000	470548.7	467458.2	12.56876	12.90986
23	5	JUVENILE	9	168.0000	138.0000	464496.7	461595.6	13.16535	13.55002
24	5	JUVENILE	9	175.0000	145.0000	458809.6	456077.8	13.82870	14.18514
25	5	JUVENILE	10	182.0000	152.0000	453449.7	450870.2	14.48736	14.81521
26	5	JUVENILE	10	189.0000	159.0000	448384.1	445942.6	15.06990	15.44025
27	5	JUVENILE	10	196.0000	166.0000	443585.7	441269.1	15.71921	16.06024
28	5	JUVENILE	10	203.0000	173.0000	439029.6	436827.1	16.36383	16.67520
29	5	JUVENILE	10	210.0000	180.0000	434695.1	432596.8	16.93234	17.28513
30	5	JUVENILE	11	217.0000	187.0000	430563.1	428560.8	17.56760	17.89000
31	5	JUVENILE	11	224.0000	194.0000	426617.7	424703.5	18.12675	18.48984
32	5	JUVENILE	11	231.0000	201.0000	422844.1	421011.3	18.75266	19.08464
33	5	JUVENILE	11	238.0000	208.0000	419229.0	417471.7	19.37389	19.67441
34	5	JUVENILE	12	245.0000	215.0000	415761.1	414073.9	19.91899	20.25912
35	5	JUVENILE	12	252.0000	222.0000	412430.1	410808.1	20.53085	20.83881
36	5	JUVENILE	12	259.0000	229.0000	409226.6	407665.1	21.06662	21.41344
37	5	JUVENILE	12	266.0000	236.0000	406141.7	404637.0	21.66911	21.98305
38	5	JUVENILE	1	273.0000	243.0000	403167.8	401716.5	22.19551	22.54761
39	5	JUVENILE	1	280.0000	250.0000	400298.1	398896.8	22.78865	23.10712
40	5	JUVENILE	1	287.0000	257.0000	397526.6	396172.0	23.37712	23.66161
41	5	JUVENILE	1	294.0000	264.0000	394846.6	393536.4	23.88947	24.21104
42	5	JUVENILE	2	301.0000	271.0000	392253.4	390984.8	24.46858	24.75545
43	5	JUVENILE	2	308.0000	278.0000	389742.0	388512.6	24.97157	25.29480
44	5	JUVENILE	2	315.0000	285.0000	387307.6	386115.4	25.54131	25.82912
45	5	JUVENILE	2	322.0000	292.0000	384946.1	383789.4	26.03494	26.35840
46	5	JUVENILE	2	329.0000	299.0000	382654.3	381530.7	26.59534	26.88266
47	5	JUVENILE	3	336.0000	306.0000	380427.6	379335.9	27.07961	27.40186
48	5	JUVENILE	3	343.0000	313.0000	378263.6	377201.9	27.63063	27.91602
49	5	JUVENILE	3	350.0000	320.0000	376158.6	375125.7	28.10555	28.42513
50	5	JUVENILE	3	357.0000	327.0000	374110.3	373104.5	28.64720	28.92921
51	5	JUVENILE	4	364.0000	334.0000	372115.6	371135.8	29.11276	29.42825
52	5	JUVENILE	4	371.0000	341.0000	370172.0	369217.2	29.64507	29.92226
53	5	JUVENILE	4	378.0000	348.0000	368277.6	367346.4	30.10126	30.41121
54	5	JUVENILE	4	385.0000	355.0000	366430.0	365521.3	30.62421	30.89513
55	5	JUVENILE	5	392.0000	362.0000	364626.7	363740.1	31.07103	31.37401
56	5	JUVENILE	5	399.0000	369.0000	362866.6	362000.8	31.58463	31.84785
57	5	JUVENILE	5	406.0000	376.0000	361147.4	360301.5	32.02209	32.31665
58	5	JUVENILE	5	413.0000	383.0000	359467.6	358640.9	32.52632	32.78041
59	5	JUVENILE	5	420.0000	390.0000	357825.6	357017.2	32.95444	33.23914

PERIOD	STAGE	NAME OF STAGE	MNTH	COST OF PERIOD	COST UP TO END OF PERIOD	AVG COST BIV IN PERIOD/DAY	COST BIVALV AT END PERIOD
1	1	CONDITION	4	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
2	2	SPAWN	5	0.7474519E-02	0.7474519E-02	0.3861864E-08	0.7969341E-08
3	3	LARVAE	5	0.4317153	0.4391898	0.1208737E-06	0.5143113E-06
4	3	LARVAE	5	2.615227	3.054416	0.7932467E-06	0.3830176E-05
5	3	LARVAE	5	7.857286	10.91170	0.2532412E-05	0.1444079E-04
6	4	SETTING	5	23.55299	34.46469	0.6411356E-05	0.4816496E-04
7	5	JUVENILE	5	18.45863	52.92332	0.3802835E-05	0.7859844E-04
8	5	JUVENILE	6	38.54652	91.46985	0.8386842E-05	0.1426773E-03
9	5	JUVENILE	6	68.17392	159.6438	0.1551432E-04	0.2594735E-03
10	5	JUVENILE	6	108.8479	268.4915	0.2573380E-04	0.4521173E-03
11	5	JUVENILE	6	161.7548	430.2461	0.3953188E-04	0.7473773E-03
12	5	JUVENILE	7	230.2422	660.4883	0.5794545E-04	0.1179565E-02
13	5	JUVENILE	7	311.4697	971.9580	0.8047794E-04	0.1779747E-02
14	5	JUVENILE	7	407.9121	1379.870	0.1079404E-03	0.2584838E-02
15	5	JUVENILE	7	520.3247	1900.195	0.1407219E-03	0.3634693E-02
16	5	JUVENILE	7	649.3794	2549.574	0.1791879E-03	0.4971918E-02
17	5	JUVENILE	8	797.7424	3347.317	0.2242634E-03	0.6645836E-02
18	5	JUVENILE	8	962.2297	4309.543	0.2752387E-03	0.8700971E-02
19	5	JUVENILE	8	1144.994	5454.535	0.3328808E-03	0.1118737E-01
20	5	JUVENILE	8	1346.435	6800.969	0.3974687E-03	0.1415722E-01
21	5	JUVENILE	9	1556.416	8357.383	0.4661228E-03	0.1764254E-01
22	5	JUVENILE	9	1794.602	10151.98	0.5448356E-03	0.2171741E-01
23	5	JUVENILE	9	2052.225	12204.21	0.6311669E-03	0.2643918E-01
24	5	JUVENILE	9	2329.478	14533.68	0.7253170E-03	0.3186667E-01
25	5	JUVENILE	10	2604.325	17138.01	0.8204796E-03	0.3801095E-01
26	5	JUVENILE	10	2918.578	20056.59	0.9298711E-03	0.4497571E-01
27	5	JUVENILE	10	3252.629	23309.21	0.1047512E-02	0.5282313E-01
28	5	JUVENILE	10	3606.510	26915.72	0.1173532E-02	0.6161642E-01
29	5	JUVENILE	10	3980.225	30895.95	0.1308051E-02	0.7141972E-01
30	5	JUVENILE	11	4331.574	35227.52	0.1437179E-02	0.8219957E-01
31	5	JUVENILE	11	4740.824	39968.34	0.1587512E-02	0.9410876E-01
32	5	JUVENILE	11	5169.527	45137.87	0.1746516E-02	0.1072130
33	5	JUVENILE	11	5617.547	50755.42	0.1914244E-02	0.1215781
34	5	JUVENILE	12	6016.117	56771.54	0.2067161E-02	0.1371048
35	5	JUVENILE	12	6496.813	63268.35	0.2250359E-02	0.1540095
36	5	JUVENILE	12	6996.082	70264.38	0.2442266E-02	0.1723580
37	5	JUVENILE	12	7513.711	77778.06	0.2642889E-02	0.1922169
38	5	JUVENILE	1	8014.852	85792.88	0.2839955E-02	0.2135657
39	5	JUVENILE	1	8566.055	94358.88	0.3057026E-02	0.2365496
40	5	JUVENILE	1	9134.746	103493.6	0.3282708E-02	0.2612339
41	5	JUVENILE	1	9720.613	113214.1	0.3516958E-02	0.2876840
42	5	JUVENILE	2	10323.35	123537.4	0.3759722E-02	0.3159648
43	5	JUVENILE	2	10942.60	134480.0	0.4010931E-02	0.3461406
44	5	JUVENILE	2	11578.04	146058.0	0.4270520E-02	0.3782755
45	5	JUVENILE	2	12229.26	158287.3	0.4538391E-02	0.4124326
46	5	JUVENILE	2	12895.91	171183.1	0.4814457E-02	0.4486746
47	5	JUVENILE	3	13577.59	184760.7	0.5098619E-02	0.4870635
48	5	JUVENILE	3	14273.87	199034.5	0.5390748E-02	0.5276604
49	5	JUVENILE	3	14984.34	214018.8	0.5690739E-02	0.5705256
50	5	JUVENILE	3	15708.59	229727.4	0.5998455E-02	0.6157185
51	5	JUVENILE	4	16570.00	246297.4	0.6361309E-02	0.6636314
52	5	JUVENILE	4	17326.07	263623.4	0.6686494E-02	0.7140064
53	5	JUVENILE	4	18094.64	281718.1	0.7019021E-02	0.7669003
54	5	JUVENILE	4	18875.22	300593.3	0.7358730E-02	0.8223686
55	5	JUVENILE	5	19898.57	320491.8	0.7796064E-02	0.8811012
56	5	JUVENILE	5	20711.18	341202.9	0.8153796E-02	0.9425476
57	5	JUVENILE	5	21534.45	362737.4	0.8518264E-02	1.006761
58	5	JUVENILE	5	22367.88	385105.3	0.8889288E-02	1.073791
59	5	JUVENILE	5	23210.93	408316.1	0.9266656E-02	1.143687

PRD	*STAGE		MNTH	COST OF INPUTS IN A PERIOD: ALGAE	COST OF INPUTS UP TO END OF A PERIOD ALGAE	QUANTITY OF INPUT USED PER PERIOD: ALGAE CELL : TERTIARY	SOLAR
2	2	SPAWN	5	0.7474519E-02	0.7474519E-02	0.7655265E+09	0.0000000E+00
3	3	LARVAE	5	0.4317153	0.4391898	0.4421547E+11	0.0000000E+00
4	3	LARVAE	5	2.615227	3.054416	0.2678467E+12	0.0000000E+00
5	3	LARVAE	5	7.857286	10.91170	0.8047285E+12	0.0000000E+00
6	4	SETTING	5	23.55299	34.46469	0.2412253E+13	0.0000000E+00
7	5	JUVENILE	5	18.45863	52.92332	0.0000000E+00	0.6950191E+13
8	5	JUVENILE	6	38.54652	91.46985	0.0000000E+00	0.1438770E+14
9	5	JUVENILE	6	68.17392	159.6438	0.0000000E+00	0.2544629E+14
10	5	JUVENILE	6	108.8479	268.4915	0.0000000E+00	0.4062808E+14
11	5	JUVENILE	6	161.7548	430.2461	0.0000000E+00	0.6037587E+14
12	5	JUVENILE	7	230.2422	660.4883	0.0000000E+00	0.8508278E+14
13	5	JUVENILE	7	311.4697	971.9580	0.0000000E+00	0.1150993E+15
14	5	JUVENILE	7	407.9121	1379.870	0.0000000E+00	0.1507383E+15
15	5	JUVENILE	7	520.3247	1900.195	0.0000000E+00	0.1922787E+15
16	5	JUVENILE	7	649.3794	2549.574	0.0000000E+00	0.2399691E+15
17	5	JUVENILE	8	797.7424	3347.317	0.0000000E+00	0.2940294E+15
18	5	JUVENILE	8	962.2297	4309.543	0.0000000E+00	0.3546556E+15
19	5	JUVENILE	8	1144.994	5454.535	0.0000000E+00	0.4220181E+15
20	5	JUVENILE	8	1346.435	6800.969	0.0000000E+00	0.4962647E+15
21	5	JUVENILE	9	1556.416	8357.383	0.0000000E+00	0.5775249E+15
22	5	JUVENILE	9	1794.602	10151.98	0.0000000E+00	0.6659065E+15
23	5	JUVENILE	9	2052.225	12204.21	0.0000000E+00	0.7615004E+15
24	5	JUVENILE	9	2329.478	14533.68	0.0000000E+00	0.8643780E+15
25	5	JUVENILE	10	2604.325	17138.01	0.0000000E+00	0.9745968E+15
26	5	JUVENILE	10	2918.578	20056.59	0.0000000E+00	0.1092197E+16
27	5	JUVENILE	10	3252.629	23309.21	0.0000000E+00	0.1217207E+16
28	5	JUVENILE	10	3606.510	26915.72	0.0000000E+00	0.1349637E+16
29	5	JUVENILE	10	3980.225	30895.95	0.0000000E+00	0.1489489E+16
30	5	JUVENILE	11	4331.574	35227.52	0.0000000E+00	0.1636750E+16
31	5	JUVENILE	11	4740.824	39968.34	0.0000000E+00	0.1791391E+16
32	5	JUVENILE	11	5169.527	45137.87	0.0000000E+00	0.1953383E+16
33	5	JUVENILE	11	5617.547	50755.42	0.0000000E+00	0.2122674E+16
34	5	JUVENILE	12	6016.117	56771.54	0.0000000E+00	0.2299207E+16
35	5	JUVENILE	12	6496.813	63268.35	0.0000000E+00	0.2482916E+16
36	5	JUVENILE	12	6996.082	70264.38	0.0000000E+00	0.2673726E+16
37	5	JUVENILE	12	7513.711	77778.06	0.0000000E+00	0.2871549E+16
38	5	JUVENILE	1	8014.852	85792.88	0.0000000E+00	0.3076293E+16
39	5	JUVENILE	1	8566.055	94358.88	0.0000000E+00	0.3287859E+16
40	5	JUVENILE	1	9134.746	103493.6	0.0000000E+00	0.3506136E+16
41	5	JUVENILE	1	9720.613	113214.1	0.0000000E+00	0.3731006E+16
42	5	JUVENILE	2	10323.35	123537.4	0.0000000E+00	0.3962351E+16
43	5	JUVENILE	2	10942.60	134480.0	0.0000000E+00	0.4200033E+16
44	5	JUVENILE	2	11578.04	146058.0	0.0000000E+00	0.4443929E+16
45	5	JUVENILE	2	12229.26	158287.3	0.0000000E+00	0.4693884E+16
46	5	JUVENILE	2	12895.91	171183.1	0.0000000E+00	0.4949761E+16
47	5	JUVENILE	3	13577.59	184760.7	0.0000000E+00	0.5211406E+16
48	5	JUVENILE	3	14273.87	199034.5	0.0000000E+00	0.5478656E+16
49	5	JUVENILE	3	14984.34	214018.8	0.0000000E+00	0.5751352E+16
50	5	JUVENILE	3	15708.59	229727.4	0.0000000E+00	0.6029335E+16
51	5	JUVENILE	4	16570.00	246297.4	0.0000000E+00	0.6312429E+16
52	5	JUVENILE	4	17326.07	263623.4	0.0000000E+00	0.6600459E+16
53	5	JUVENILE	4	18094.64	281718.1	0.0000000E+00	0.6893246E+16
54	5	JUVENILE	4	18875.22	300593.3	0.0000000E+00	0.7190613E+16
55	5	JUVENILE	5	19898.57	320491.8	0.0000000E+00	0.7492369E+16
56	5	JUVENILE	5	20711.18	341202.9	0.0000000E+00	0.7798338E+16
57	5	JUVENILE	5	21534.45	362737.4	0.0000000E+00	0.8108318E+16
58	5	JUVENILE	5	22367.88	385105.3	0.0000000E+00	0.8422130E+16
59	5	JUVENILE	5	23210.93	408316.1	0.0000000E+00	0.8739563E+16