

FOR REFERENCE

NOT TO BE KEPT IN READING ROOM

A COMPUTER-AIDED
PRODUCTION PLANNING SYSTEM
FOR A
FERTILIZER MANUFACTURING COMPANY

BY

FEZA AKSU

B.S., in Industrial Engineering, Boğaziçi University, 1984

Submitted to the Institute for Graduate Studies in
Social Sciences in partial fulfillment of the
requirements for the degree of
Master of Arts

in

Business Administration

Bogazici University Library



Boğaziçi University

1987

A COMPUTER-AIDED
PRODUCTION PLANNING SYSTEM
FOR A
FERTILIZER MANUFACTURING COMPANY

APPROVED BY :

Doç. Dr. Ceyhan Uyar

.....*Ceyhan Uyar*.....

Sibel Tanberk

.....*Sibel Tanberk*.....

Doç. Dr. Seyhan Tuğcu

.....*Seyhan Tuğcu*.....

DATE OF APPROVAL : March, 23rd 1987

200841



TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
ABSTRACT	v
ÖZET	vi
LIST OF TABLES AND FIGURES	vii
LIST OF ABBREVIATIONS	viii
I. INTRODUCTION	1
II. CHEMICAL FERTILIZERS	4
2.1 PLANT NUTRIENTS AND FERTILIZERS	4
2.2 FERTILIZER TYPES	6
2.3 COMPOUND FERTILIZERS	11
III. WORLD FERTILIZER SECTOR	12
IV. TURKISH FERTILIZER SECTOR	17
4.1 BRIEF OVERVIEW	17
4.2 MONOPOLISTIC STAGE	28
4.3 LIBERAL MARKET STAGE	30
V. HIGHLIGHTS ON TOROS	31
5.1 BRIEF OVERVIEW	31
5.2 PRODUCT MIX	32
5.3 RELATED OFFSITES	33
5.4 PRODUCTION PROCESSES	34
5.5 AVAILABILITY OF RAW MATERIALS	35
VI. PRODUCTION CONTROL SYSTEM FOR TOROS	39
6.1 ABC ANALYSIS	42
6.2 INVENTORY PLANNING	43
6.3 BILL OF MATERIALS	53
6.4 MATERIAL REQUIREMENTS PLANNING	54
6.5 CAPACITY REQUIREMENTS PLANNING	57
VII. SUMMARY OF FINDINGS	59
7.1 ABC ANALYSIS	60
7.2 INVENTORY PLANNING	62
7.3 BILL OF MATERIALS	63
7.4 MATERIAL REQUIREMENTS PLANNING	64
7.5 CAPACITY REQUIREMENTS PLANNING	65
APPENDIX A : FILE NETWORK DIAGRAM	70
APPENDIX B : INPUT FILES	80
APPENDIX C : OUTPUT FILES	93
APPENDIX D : SOURCE FILES	104
REFERENCES NOT CITED	226

ACKNOWLEDGEMENTS

I would like to express my gratitudes to my thesis supervisor Doç. Dr. Ceyhan Uyar, and to the members of my thesis committee Sibel Tanberk and Doç. Dr. Seyhan Tuğcu for their most valuable contributions and finally to the directors of Toros, without whose understanding this study could not have been carried through.

v.

A COMPUTER-AIDED
PRODUCTION PLANNING SYSTEM
FOR A
FERTILIZER MANUFACTURING COMPANY

This study aims to design a specific Computer-Aided Production Planning System for a private chemical fertilizer manufacturing company in a recently liberalized market, within the bounds of the environmental and organizational conditions.

The respective computer package has been designed as an integrated whole to achieve maximum flexibility and ease of application. The package consists of three main modules : Inventory Planning to maintain inventory investment at an optimum level; Material Requirements Planning to secure time phased availability of the inventory items and finally Capacity Requirements Planning to match the production schedules with the productive capability of the production centers. The framework includes two supplementary modules, namely ABC Analysis and Bill of Materials and read/write programs for the data files.

GÜBRE ÜRETİCİSİ BİR KURULUS İÇİN
BİLGİSAYARA DAYALI
ÜRETİM PLANLAMASI SİSTEMİ

Bu çalışma, serbest pazarlama sistemine geçiş aşamasını yaşamakta olan kimyevi gübre sektöründe üretici firma olarak hizmet veren özel bir kuruluş için, bünyesel ve çevresel şartlar çerçevesi içerisinde, Üretim Planlaması Sistemi kurmayı amaçlamaktadır.

Bu amaçla geliştirilen bilgisayar modeli uygulama ve kullanım esnekliği faktörleri gözönüne alınarak küçük modüllerden oluşan entegre bir bütün olarak dizayn edilmiştir. Optimal düzeyde stok yatırımını planlayan Stok Kontrol Programı (Inventory Planning), hammadde temininin ve üretimin zamanlamasını planlayan Malzeme ihtiyaçları Planlaması (MRP) ve planlanan üretimi üretim tesislerinin kapasiteleri ile karşılaştıran Kapasite ihtiyaçları Planlaması (CRP) geliştirilen sistemi oluşturan üç ana modüldür. Üretim Planlaması Sisteminin ana yapısı içinde ABC Analizi (ABC Analysis) ve Malzeme Girdileri (Bill of Materilas) programları ve veri okutma/yazdırma programları da yer almaktadır.

LIST OF TABLES AND FIGURES

TABLE 2.1	Nutrient Composition of some Fertilizer Materials
FIGURE 2.1	Downstream Nitrogen Fertilizers From Anhydrous Ammonia
FIGURE 2.2	Downstream Phosphate Fertilizers from Phosphate Rock
TABLE 3.1	World Fertilizer Production (1985)
TABLE 3.2	World Fertilizer Consumption (1985)
FIGURE 3.1	Fertilizer Use per Hectare of Arable Land
TABLE 4.1	Turkish Fertilizer Production Capacity (1986)
TABLE 4.2	Turkish Fertilizer Consumption by Type (1985)
TABLE 4.3	Production-Imports-Exports-Consumption of Fertilizers in Turkey (1986)
TABLE 4.4	Turkish Fertilizer Consumption by Crops (1984)
TABLE 4.5	Turkish Fertilizer Consumption by Regions (1985)

LIST OF ABBREVIATIONS

- BOM : Bill Of Materials
- CRP : Capacity Requirements Planning
- MRP : Material Requirements Planning
- ŞEKER : The Sugar Factories Company
- TOROS : Toros Fertilizer and Chemical
Industry Co., Inc.
- TZDK : The Agricultural Supply Organization
(Türkiye Ziraî Donatım Kurumu)

I. INTRODUCTION

Toros Fertilizer and Chemical Industry Co., Inc., which had been established with the aim of serving the agricultural sector in the best possible way, commenced production of monoammonium phosphate and high grade compound fertilizers in 1981. After three successful years of operation, the company doubled its production, packaging, handling and storage capacities and included diammonium phosphate in its product mix.

Fertilizers play a key role in raising agricultural productivity. Unlike the case in most other developing countries, fertilizer use has become fairly widespread in Turkey over the last 20 years. Fertilizer procurement and distribution have been a monopoly of two public agencies, the Agricultural Supply Organization (TZDK) and the Sugar Factories Company (SEKER) until 1986. TZDK handled total imports of finished fertilizers and over 90 per cent of the domestic procurement and distribution. The balance was handled by SEKER. Both the ex-factory and retail prices of fertilizers were determined by the Government through semi-annual decrees on a negotiated 'cost plus' method for each fertilizer type. The Government's policy then, was to protect the domestic fertilizer industry through a system of

administered ex-factory prices set above the border equivalent of international prices. Domestic manufacturers were assured of the sale of their total production, and they operated under certainty as to adequate compensation for increases in costs of imported materials resulting from exchange rate fluctuations. This system, therefore, did not provide any incentive to reduce production costs.

The year 1986, witnessed very significant changes in the fertilizer sector, as the monopoly of TZDK and SEKER in fertilizer procurement and distribution came to an end, and the era of liberal fertilizer market begun. Turkish fertilizer producers are now permitted to export, import and market fertilizers throughout the country at independently determined prices.

Following the liberalization of the fertilizer distribution system, Toros suddenly found itself in a totally competitive market. Whereas in the past all the production of the company was purchased by TZDK and SEKER, there are no assurance now as to this respect. The domestic demand is now mostly met by purchases through the bids opened by the public agricultural agencies. Thus, there exists a tough competition between Toros and other compound fertilizer and diammonium phosphate suppliers. As the chances are always there, that Toros will not be able to sell its total output domestically, the company seeks export possibilities. Yet, international bids are even more

competitive and the export prices are much lower than the domestic prices. Thus, cost reduction policy is a must for Toros's survival. On the other hand, in case that Toros is entitled to the task of supplying fertilizers throughout Turkey, scheduling becomes dramatically important. The main objective in fertilizer marketing should be considered as securing the availability of fertilizer supplies in the right combination, at the right place and at the right time. The essence of the problem of developing a production program is to supply the finished products in the quantities specified in the sales program, to have them available before the requirement dates, and to so arrange production that labour, plant and capital will be used in the most efficient way possible. The result must be a compromise; the problem is to find the best compromise.

As a conclusion, Toros has no other choice but to adapt itself to the gradual change from the previous passive marketing approach to a progressively aggressive marketing approach in the distribution of fertilizers. If it intends to maintain its strong position in the fertilizer market, the company is in need of implementing a computerized production planning system designed to minimize the total cost of meeting a stated future demand on time within the constraints of the existing environmental and organizational conditions.

II. CHEMICAL FERTILIZERS

2.1 Plant Nutrients and Fertilizers

If we accept that the major challenge for the future is to feed the world, then the next major challenge is how to obtain higher agricultural productivity from essentially the same agricultural land area. Thus the most important single objective in farming becomes to increase the crop yield and the agricultural productivity.

There are 16 nutrients known to be essential for plant growth. The three major constituents of plant structure, oxygen, hydrogen and carbon, because of their abundance and ease of availability from water and air, are not considered as commercial fertilizer constituents. Plant nutrients are grouped into three categories as 'primary', 'secondary' and 'micro' nutrients. Nitrogen, phosphorus and potassium are primary nutrients, so called because they are required in relatively large quantities. With oxygen, hydrogen and carbon, these primary nutrients are essential in the formation of the plant structure. Secondary nutrients and micronutrients are generally required in smaller quantities than the primary nutrients. While some of the secondary nutrients and micronutrients may join some of the primary nutrients as constituents of the plant body, others merely serve as catalysts of necessary chemical reactions.

Substances containing plant nutrients occur naturally or are compounded and produced as commercial fertilizers. The major source of plant nutrients is the mineral supply of the soil. As a result of farming, the organic matter content of the soil decreases to a level at which as much nutrient should be added as has been removed by the biomass. The use of fertilizers is a key factor for restoring and also increasing the fertility of the soil and thus raising agricultural production.

2.2 Fertilizer Types

The term fertilizers corresponds to chemical and mineral products containing one or more of the plant nutrients. Fertilizers differ in their nutrient content, both as to kind and percentage, and they are designed for certain uses. Although each nutrient element is credited with specific functions in the plant, it is important to realize how intricately balanced and interrelated the entire plant growth process is.

Nitrogen (N) is the key element or nutrient in plant growth. A proper balance and adequate supply of other nutrients must generally be maintained, but the amount of nitrogen should be adjusted for the desired growth response. Variation in available nitrogen for plant use determines the greenness of the leaves, ability of the plant to recover from damage or stress, and the quantity of clippings the grass produces. The primary nitrogen fertilizers are ammonium nitrate, ammonium sulfate, nitrogen solutions, urea and urea formaldehyde.(Fig.2.1)

Phosphorus (P) has been labeled the workhorse of the nutrition team. Phosphorus provides the plant with a mechanism for using and transforming energy. Single and triple superphosphate, and diammonium phosphate are the main phosphate fertilizers.(Fig.2.2)

Potassium (K) is used by the plant in relatively large quantities. Potassium favors the development of cell walls, thus making the plant more resistant to heat, cold and frost as well as to disease attack. Potassium's role is that of a regulator of plant processes; it influences the performance of at least 46 enzymes, and controls the uptake of some nutrients. The main phosphate fertilizers are potassium chloride, potassium magnesium sulfate and potassium nitrate.

The nutrient contents of major fertilizers are shown on Table 2.1.

FIGURE 2.1

DOWNSTREAM NITROGEN FERTILIZERS
FROM ANHYDROUS AMMONIA

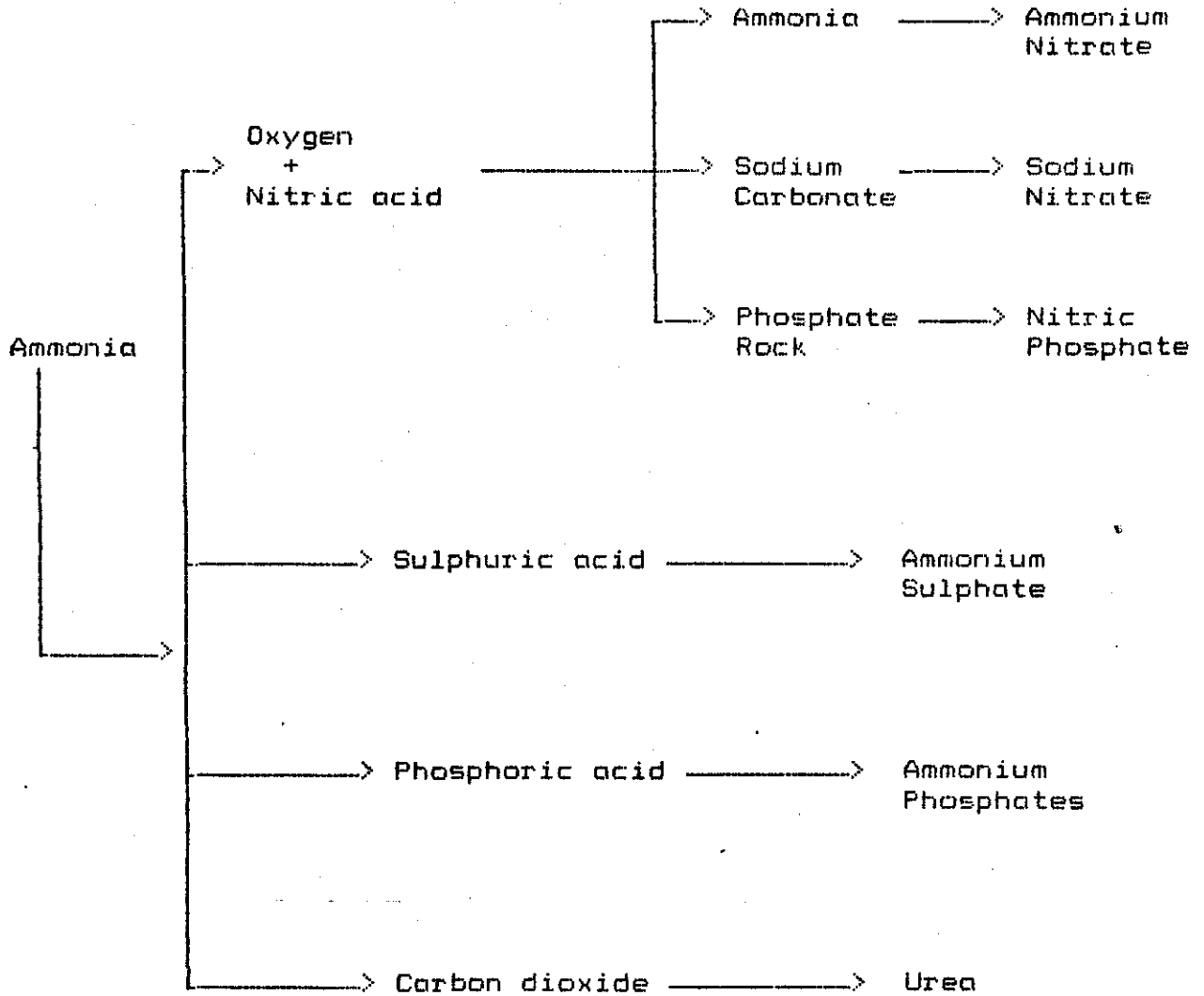


FIGURE 2.2

DOWNSTREAM PHOSPHATE FERTILIZERS
FROM PHOSPHATE ROCK

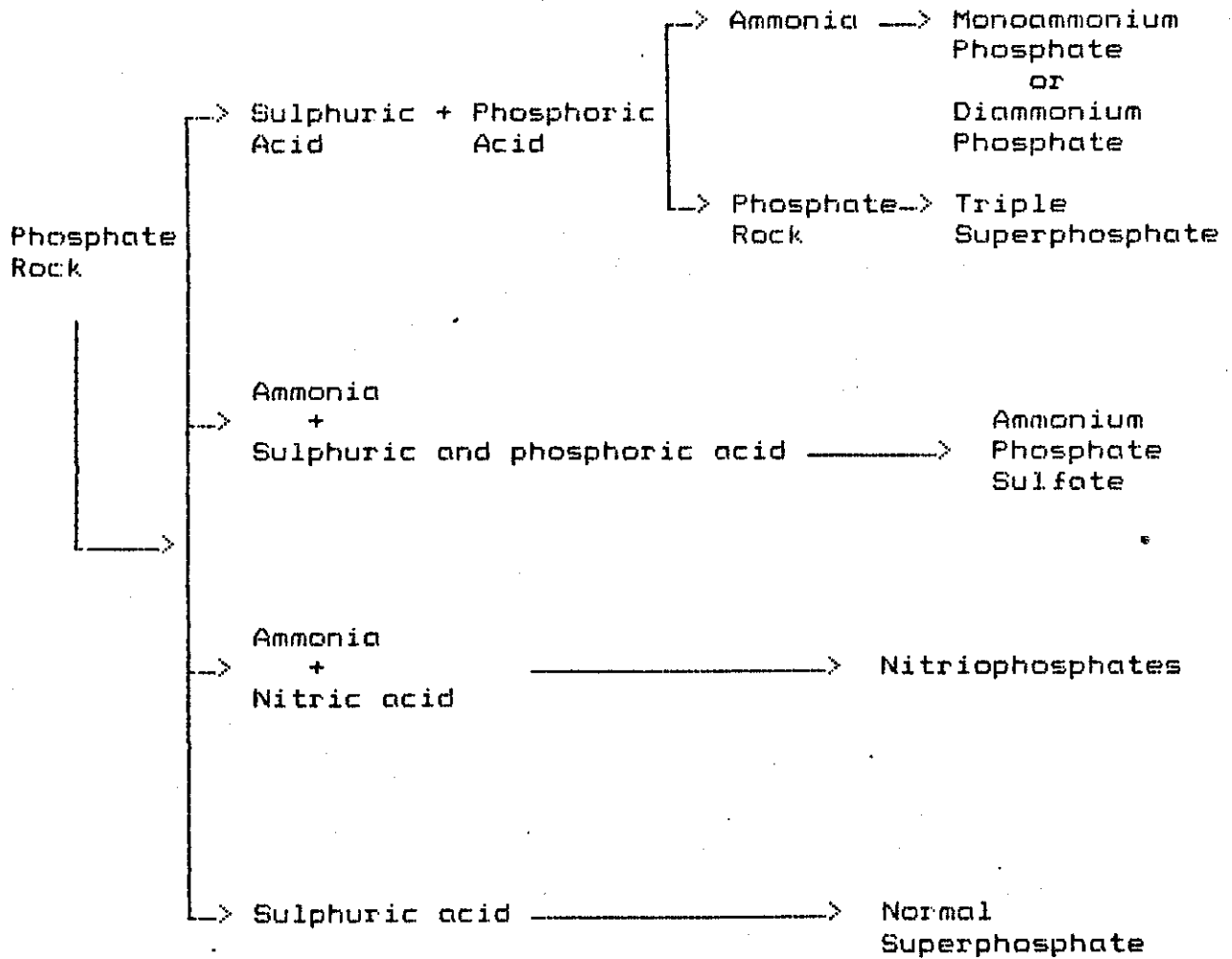


TABLE 2.1
NUTRIENT COMPOSITION OF
SOME FERTILIZER MATERIALS

<u>FERTILIZER MATERIAL</u>	<u>% OF NITROGEN</u>	<u>% OF PHOSPHATE</u>	<u>% OF POTASH</u>
Ammonium Nitrate	33.5	-	-
Ammonium Sulphate	21	-	-
Nitrogen Solutions	21-49	-	-
Urea	46	-	-
Urea-form	38	-	-
Super-Phosphate	-	18-20	-
Concentrated Superphos	-	42-50	-
Potassium Chloride	-	-	60-62
Potassium Magnesium Sulphate	-	-	22
Diammonium Phosphate	16-21	48-53	-
Potassium Nitrate	13	-	44

2.3 Compound Fertilizers

There are two fertilizer categories which are as :

- (a) Straight Fertilizers: Fertilizers containing only one of the three main plant nutrients N, P or K.
- (b) Compound Fertilizers : Fertilizers containing more than one of the three plant nutrients, and they are referred to as NPK types; with 'N' being the symbol for nitrogen, 'P' being the symbol for phosphorus and 'K' being the symbol for potassium.

The main characteristics of the NPK fertilizers is that nutrients are available in various grades and chemical forms suitable for any crop, climate and soil. Fertilizer grade is the minimum guarantee of available plant food expressed in terms of total nitrogen, available phosphoric acid or phosphorous, and soluble potash or potassium. For example, a grade of 18-24-12 signifies that the product is guaranteed to contain 18 percent of its weight as nitrogen, 24 percent as phosphate and 12 percent as potash.

III. WORLD FERTILIZER SECTOR

Although the world is clearly not running out of land, there are serious problems about its availability and suitability for agriculture. The major problem faced is not the limitation of land but low yields. Therefore, yield increasing technologies are to be the primary source of growth in food production. Fertilizer use is one of the key factors in yield increases.

The intensive use of fertilizers as a mean of increasing agricultural production has become a general practice in the world. (Fig.3.1) Arable area in use as a percentage of potential area is expected to reach 50 per cent for the developed countries by year 2000.

The cyclical pattern of the fertilizer industry is characterized by long periods of oversupply followed by brief periods of shortages. To supply the growing demand between 1975 and 1981, additional capacities were built all over the world, although much more extensively in the developing countries where several new fertilizer production centers have emerged. Except for potash, this transformation has changed the pattern of capacity share between the developed and the developing countries. In spite of the severe economic depression of 1980 to 1983 and the decline in demand for fertilizers in the developed countries, the overall demand for fertilizers is found to increase at an

annual rate of four to eight million tonnes, of which three to four million tonnes are for nitrogenous fertilizers and the balance for phosphate or potash fertilizers. The largest portion of the increase in demand can be attributed to the developing countries such as the sub-continent of Asia and Central and South America. In Africa, where the food deficit is equally great, the demand for fertilizers remains at the lowest level. The inequality in the use of fertilizers between various regions indicates the existence of large potential demand and the possibility of greatly increasing supply.

In nutrient terms about 139.2 million tons were produced in 1985, of which 74.7 million tons was of nitrogen, 35.6 million tons was of phosphate, and the remaining was of potash. Europa is the leading regional nutrient producer; next come Asia and North America (Table 3.1). World fertilizer consumption has reached to 130.6 million tons in plant nutrient terms in 1985, equivalent to 70.6 million tons of nitrogen, 34.1 million tons of phosphate and 25.9 tons of potash nutrients. The USSR leads the world in plant nutrient consumption. (Table 3.2) In 1981/82, the developing countries consumed more than 38 million tons of fertilizer nutrients, amounting to one third of the world total consumption. The developed world and more particularly North America and the USSR have a dominant share in the supply of all fertilizer raw materials.

FIG. 3.1

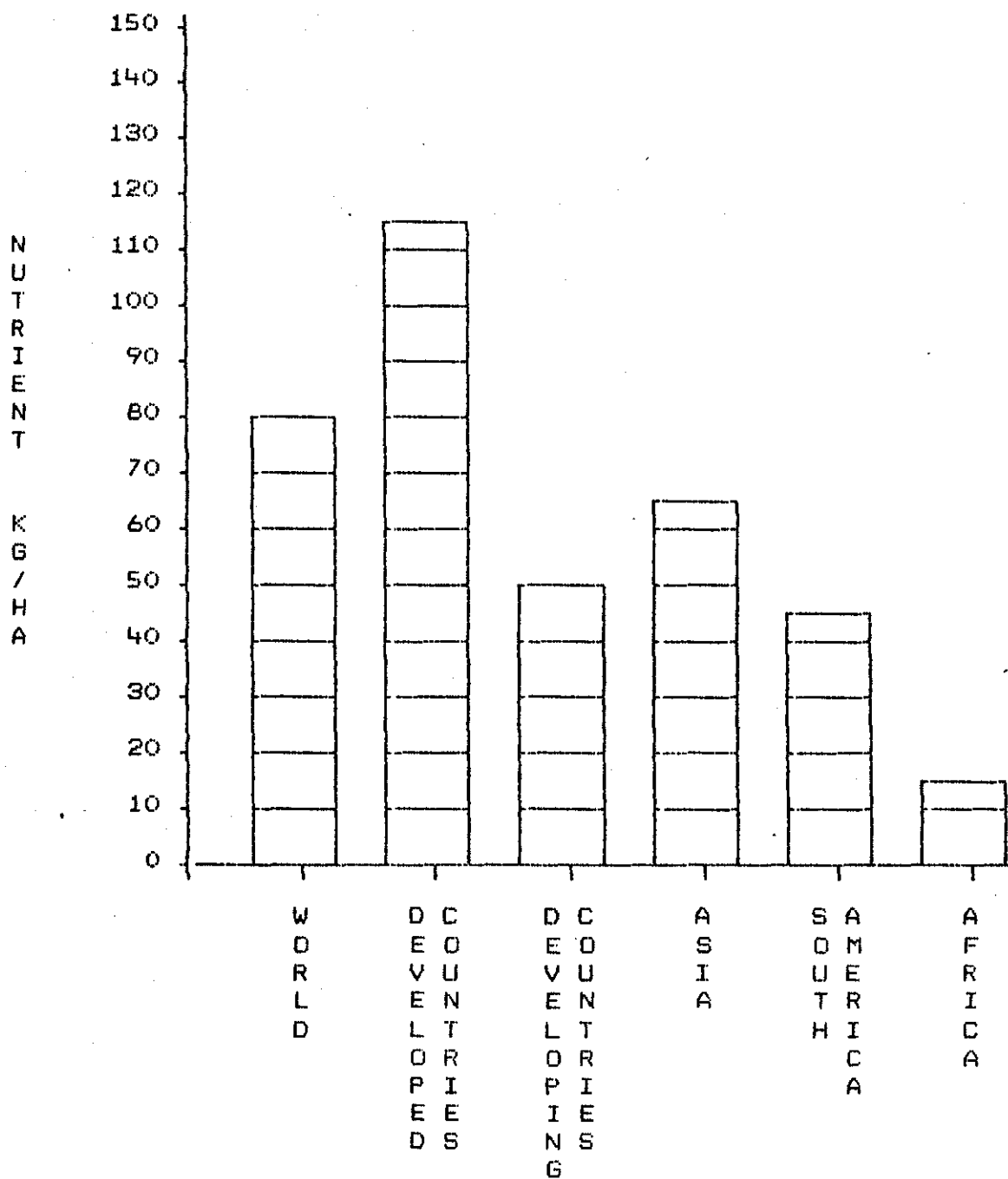
FERTILIZER USE PER HECTARE
OF ARABLE LAND

TABLE 3.1

WORLD FERTILIZER PRODUCTION
1985

(in million tons of plant nutrients)

<u>REGIONS</u>	<u>NITROGEN</u>	<u>PHOSPHATE</u>	<u>POTASH</u>	<u>TOTAL</u>
W. Europe	12.1	5.4	5.5	23.0
Africa	1.8	1.6	-	3.4
N. America	14.2	9.0	8.7	31.9
S. America	2.4	1.8	-	4.2
Asia	10.5	3.9	1.5	15.9
E. Europe	7.3	3.2	3.5	14.0
Russia	13.1	7.1	9.7	29.9
Others	13.3	3.6	-	16.9
TOTAL	74.7	35.6	28.9	139.2

TABLE 3.2

WORLD FERTILIZER CONSUMPTION
1985

(in million tons of plant nutrients)

<u>REGIONS</u>	<u>NITROGEN</u>	<u>PHOSPHATE</u>	<u>POTASH</u>	<u>TOTAL</u>
W. Europa	11.0	5.6	5.3	21.9
Africa	1.9	1.2	0.4	3.5
N. America	11.7	5.0	5.4	22.1
S. America	2.9	2.4	1.5	6.8
Asia	11.2	4.9	2.5	18.6
E. Europa	5.1	3.2	3.3	11.6
Russia	10.3	6.6	6.2	23.1
Others	16.5	5.2	1.3	23.0
TOTAL	70.6	34.1	25.9	130.6

IV. TURKISH FERTILIZER SECTOR

4.1 Brief Overview

To maintain a strong agricultural production base to support its population, which is increasing at an annual rate of 2.2 per cent, is crucial to Turkey's development.

Turkish fertilizer production started, on a small scale at Karabuk Iron and Steel Works, with the production of ammonium sulphate in 1939; and single superphosphate as by-product in 1944. The first independent fertilizer plant started production in 1954; and heavy investments have been observed during the second and third five-year plan periods. Today there are seven companies producing fertilizers exclusively. Of these, two - Tügsaş and İgşaş - are in the public sector and operate eight plants; two - Akdeniz Gübre and Gübre Fabrikaları - are in the joint sector and operate five plants; and three - Bağfaş, Ege Gübre and Toros are in the private sector and operate six plants. In addition, three public sector companies - Petkim, Türkiye Demir-Çelik and Ereğli Demir-Çelik - are producing fertilizers as by-products.

The installed fertilizer production capacity in Turkey was .3 million product tons in 1963; and has reached to over 5.4 million product tons, equivalent to 1.1 million tons of nitrogen and .9 million tons of phosphate nutrients

in 1986. (Table 4.1) The public sector companies account for 44 per cent of total fertilizer production capacity, the joint sector companies accounts for 26 per cent and the private sector companies account for the remaining 30 per cent.

Consumption of chemical fertilizers in Turkey on a large scale started in the late 1960's. During the following decade, fertilizer consumption witnessed continuous and rapid increase. During 1986, close to 4.0 million tons of fertilizer material, amounting to approximately 1.6 million tons of major fertilizer nutrients, were consumed. Of the total plant nutrients consumption, 1.0 million tons were of nitrogen and 0.6 million tons were of phosphate nutrients. Only negligible quantities of potash products were consumed, since Turkish soils are rich in potash and do not require additional applications.

Fertilizer consumption in Turkey is predominantly met from domestic production. For most products and seasons, there appears to be sufficient seasonal demand utilizing almost all of the domestic production with additional demand in peak seasons met by imports, mostly of nitrogenous fertilizer products and diammonium phosphate. (Table 4.2, Table 4.3) In 1986, the domestic fertilizer production amounted to 1.4 million tons of plant nutrients, equivalent to some 80 per cent of the total fertilizer consumption in the same year. This increase is largely due to of improved

availability of imported raw and intermediate materials. The import content of domestic production is substantial, since the domestic industry is largely based on imported raw or intermediate materials.

Unlike the case of most other developing countries, fertilizer use has become fairly widespread in Turkey. The capacity utilization rate in the fertilizer industry in Turkey was down to 36.5 per cent in 1978, and it achieved the highest point of 78 per cent in nutrient capacity in 1984.

The medium-term increase in the fertilizer demand in Turkey is projected to be very modest because of the already high proportion of urea cultivated with fertilizers and the relatively high levels of application per unit area. In 1980, fertilizers were applied to 70 per cent of all area cultivated.

Almost two-thirds of the fertilizers consumed in Turkey are for cotton, tobacco, sugarbeet, sunflower, fruits and vegetables. Wheat still continues to be the single largest consumer of fertilizers. (Table 4.4) Survey and trials on consumption of major nutrients and corresponding yields reveal that the existing application levels per unit area of nitrogen and phosphate are, for most crops, fairly close to the financially optimum level.

Despite the already high proportion of area cultivated with fertilizers and the relatively high levels of application per unit area in relation to the optimum, the average fertilizer consumption per unit area of arable land appears to be relatively low in Turkey as compared to some of the more developed countries.

In Turkey, consumption has developed with greater intensity along the coast and in the hinterland of the ports. (Table 4.5) The southern, western and northwest areas have developed fairly well, leaving the eastern part of the country far behind in fertilizer consumption. The application rates per unit area currently found in the Coastal and Central Plateau areas, are already very high by international standards. In some instances, rates should actually decrease, as better information is conveyed to the farmers with respect to fertilizer utilization.

Fertilizer consumption follows a seasonal pattern. Total offtake shows a peak at the beginning of the year, in February, March and April and a second peak in October and November. It should be noted, however, that this total is made up of two opposed seasonal patterns. Nitrogen fertilizers show a strong peak in February, March and April, while the phosphate fertilizers show a very strong peak in the August - November period. The annual offtake pattern represents a smoothed composite made up of the individual product variations. Further smoothing usually arises from

variations in offtake pattern in different agricultural regions of the country.

Another noticeable feature of the Turkish fertilizer consumption is that the range of products which are made available to farmers is fairly narrow and standardized throughout the country. Low-analysis fertilizers are predominant, particularly in the case of nitrogenous products, despite the savings in transportation and field application costs which can be made at the farm level through the use of high-analysis products. Similarly, phosphate is offered to farmers largely in the form of triple superphosphate, despite similar advantages offered by diammonium phosphate and compound fertilizers. The major five basic fertilizer types - calcium ammonium nitrate, ammonium sulphate, urea, 20.20.0, diammonium phosphate and single and triple superphosphate account for 98 per cent of all fertilizers consumed in Turkey.

Further increases in fertilizer consumption are to be expected essentially from :

- a) Increase in fertilizer use per unit area where the actual application is below levels which maximize farmers' profits.
- b) Use of fertilizers by farmers presently not using.
- c) Increase in irrigated area, reduction in fallow land and multiple cropping.
- d) Changes in the cropping pattern in favor of fertilizer-

intensive crops such as fruits and vegetables.

- e) Changes in the prices of fertilizers relative to output prices.
- f) Improvements in the technological package to increase yield response to fertilizers use which will lead to higher levels of fertilizer application per unit area.

Any increase in the real price of fertilizers is likely to affect demand negatively, unless it is accompanied by a corresponding increase in the agricultural production support prices. While to many observers current fertilizer prices in international trade seem relatively depressed in dollar terms, with the current strength of the US dollar, the prices of raw materials, intermediates and products in local currencies are increasing quite dramatically since they are all traded in dollars. If fertilizer material becomes too expensive, then price resistance must obviously be expected, either at or before the point at which cost/benefit ratios are unattractive, to reduce consumption, probably in those areas where it is most needed, and thereby make the basic challenge of feeding the worlds' population even more difficult.

TABLE 4.1
 TURKISH FERTILIZER
 PRODUCTION CAPACITY
 1986

FERTILIZER TYPE	NUMBER OF PRODUCTION UNITS	PRODUCTION CAPACITY (000 tons)	SHARE IN TOTAL CAPACITY (%)
Ammonium Sulphate	6	416.6	7.7
Calcium Ammonium Nitrate	4	1,576.5	29.2
Urea	1	511.5	9.5
Single Superphosphate	2	242.0	4.5
Triple Superphosphate	4	750.0	13.8
Diammonium Phosphate	3	540.7	10.0
Compounds	5	1,366.5	25.3
TOTAL	25	5,403.8	100.0
PLANT NUTRIENTS			
Nitrogen		1,100.8	55.4
Phosphate		885.7	44.6
Potash		-	-
TOTAL		1,986.5	100.0

TABLE 4.2
 TURKISH FERTILIZER CONSUMPTION
 BY TYPE
 1985

(in 000 tons of product)

FERTILIZER TYPE	CONSUMPTION	SHARE IN TOTAL (%)
Ammonium Sulphate	427	10.7
Calcium Ammonium Nitrate	1,498	37.5
Urea	409	10.2
Triple Superphosphate	299	7.5
Diammonium Phosphate	367	9.2
Compounds	924	23.1
Others	74	1.8
TOTAL	3,998	100.0

TABLE 4.3

PRODUCTION-IMPORTS-EXPORTS-CONSUMPTION
OF FERTILIZERS IN TURKEY
(1986)

(in 000 tons of plant nutrients)

<u>PLANT NUTRIENTS</u>	<u>PRODUCTION</u>	<u>IMPORTS</u>	<u>EXPORTS</u>	<u>CONSUMPTION</u>
Nitrogen	745.0	211.2	51.0	1,024.0
Phosphate	650.0	80.0	198.9	542.1
Potash	30.8	1.6	4.0	42.5
TOTAL	1,425.8	292.8	484.5	1,608.6

TABLE 4.4

TURKISH FERTILIZER CONSUMPTION
BY CROPS
1984

(in 000 tons of plant nutrients)

<u>CROPS</u>	<u>CONSUMPTION</u>
Wheat	669.2
Barley	135.0
TOTAL CEREALS	874.2
Cotton	120.0
Sugarbeet	100.0
Sunflower	100.0
Fruits	160.0
Vegetables	75.0
Others	100.0
TOTAL	1,529.2

TABLE 4.5

TURKISH FERTILIZER CONSUMPTION
BY REGIONS
1985

(in 000 tons of plant nutrients)

<u>REGIONS</u>	<u>FERTILIZER CONSUMPTION</u>	<u>SHARE IN TOTAL (%)</u>
Midnorth	237	16.6
Aegean	208	14.6
Marmara	239	16.7
Mediterranean	249	17.4
Northeast	26	1.8
Southeast	96	6.7
Blacksea	118	8.3
Mideast	83	5.8
Midsouth	171	12.1
TOTAL	1,427	100.0

4.2 Monopolistic Stage

Fertilizer procurement and distribution have been a monopoly of two public agencies, the Agricultural Supply Organization (TZDK) and the Sugar Factories Company (SEKER) until 1986. During this period TZDK handled all imports of finished fertilizers and over 90 per cent of the domestic procurement and distribution. While TZDK catered for all the regions, SEKER being an organization relating to sugar factories, distributed fertilizers mostly to farmers growing sugarbeet. Under this distribution system, TZDK was responsible for the organization and financing of fertilizer procurement, transportation and delivery down to the final retail sales outlets.

Under this system, both the ex-factory and the retail prices of fertilizers were determined by the Government through semi-annual decrees. The Government's policy since the late 1960's had been to (a) protect the domestic industry through a system of administered ex-factory prices set above the border equivalent of international prices; and (b) set uniform retail prices throughout the country at lower levels in order to encourage the use of fertilizers. Both TZDK and SEKER paid the domestic manufacturers on the basis of ex-factory price of each product fixed by the Government. Beginning in 1980 and

in line with the outward oriented policies of government, ex-factory pricing formulas were adopted which linked domestic prices with reference c.i.f. import prices. In this way, domestic manufacturers now operated under more certainty as to the ex-factory prices and were assured of adequate compensation for increases in the costs of imported materials resulting from exchange rate fluctuations. The system therefore, did not provide any incentive to reduce costs.

The virtual monopoly of TZDK as a supplier of fertilizers had a negative effect on the quality of the service provided to farmers. As TZDK saw its role as being essentially logistical, it had limited its distribution activities to a relatively small number of basic fertilizer types, already well known to farmers. At the same time, domestic manufacturers had been shielded from their own market since TZDK was the sole customer, automatically guaranteeing the purchase of factory outputs. There was no incentive in the system to seek customer satisfaction through the provision of a wider range of products, and the promotion of new products.

4.3 Liberal Market Stage

The year 1986 witnessed very significant changes in the Turkish fertilizer sector. From the beginning of the second half of 1986, the monopoly of two public agencies, TZDK and SEKER, in fertilizer procurement and distribution came to an end, and the era of liberal fertilizer market begun. The main items of the market liberalization legislation are as follows :

- (a) besides TZDK and SEKER, Turkish fertilizer producers and their affiliated marketing companies are permitted to market fertilizers throughout the country;
- (b) the fertilizer marketers are let free to determine the wholesale and retail prices;
- (c) the Government will continue to subsidize fertilizers at pre-determined rates;
- (d) fertilizer imports and exports are liberalized.

The liberalization of marketing of domestically produced and imported material should provide economic motivation to the entire system. It should improve overall availability of fertilizers, introduce further flexibility in business transactions, enable use of marketing tools and techniques, result in extensive advertisement and activate extension, leading to market penetration and conversion of potential into actual demand.

HIGHLIGHTS ON TOROS

5.1 Brief Overview

With the impetus of ever growing need for fertilizers, Tekfen Holding Company, has decided to initiate a compound fertilizer project, and Toros Fertilizer and Chemical Co., Inc. has thus been established with the participation of Agricultural Bank of Turkey and Agricultural Credit Cooperatives.

The plant site has been carefully selected in the south of Turkey, in Iskenderun Gulf enabling easy access to a network of land and sea transportation as well as having a hinterland of extensive farming. The production plant and related offsites have commenced operation in 1981 and have since then shown a remarkable performance.

5.2 The Product Mix

Initially, the Toros's complex included a NPK (compound fertilizer) unit of 330,000 mtpy and a MAP (monoammonium phosphate) unit of 130,000 mtpy, as well as related offsites. By the completion of the First Expansion Project in 1986, the new DAP/NP production unit commenced operation. Thus compound fertilizer production capacity reached to 660,000 mtpy. In addition, the flexibility to produce DAP (diammonium phosphate) at a rate of 200,000 mtpy were attained in this new unit.

Toros produces all grades of high-analysis compound fertilizers, the main characteristics of which are that the three primary nutrients, nitrogen, phosphate and potash, are available in various grades and chemical forms suitable for any crop, climate and soil. High-analysis fertilizers are cost effective right from the production stage up to and including the actual application stage in the farmers' fields. Although it is obviously advantageous to use high-analysis fertilizers, the change-over from the traditionally known fertilizers to higher analysis fertilizers has been slow in Turkey.

5.3 Related Offsites

5.3.1 Bagging Terminal

Besides for the plant's own production, Toros provides bagging services for various imported fertilizers and for transshipments to neighbouring countries. In parallel to the expansion in production capacity, the capacity of the bagging terminal has been increased to 1,320,000 mtpy.

5.3.2 Storage Facilities

The storage facilities within the Toros premises are as follows:

(A) Solid Storage :

- 17,000 mtons bagged fertilizer
- 35,000 mtons bulk fertilizer
- 32,000 mtons bulk raw material
- 40,000 mtons bulk sulphur

(B) Liquid Storage :

- 15,000 tons ammonia
- 18,000 tons phosphoric acid
- 3,000 mtons sulphuric acid
- 115,000 cu.mt. petroleum products

5.4 Production Processes

5.4.1 NPK Process

The NPK plant is based on the FISON'S process, which has been developed to make high concentration NP and NPK granular fertilizers with high quality and good storage properties. MAP, urea, filler and potash are mixed in right proportions for the production of a specific grade and granulated in the presence of ammonia and sulphuric acid.

5.4.2 MAP Process

The MAP unit is based on the FISON'S MINIFOS process which produces a powdered monoammonium phosphate by a prilling technique. MAP is produced by reacting phosphoric acid with ammonia in a reactor under a slight pressure.

5.4.3 DAP Process

The DAP unit is mostly utilized for the production of diammonium phosphate, but it also has the flexibility of producing compound fertilizers. The manufacture of diammonium phosphate is performed by neutralizing phosphoric acid with ammonia.

5.5 Raw Materials Availability

The four basic raw material sources in fertilizer production are hydrocarbons, phosphate rock, potash salts and sulphur. Toros utilizes these raw materials in processed forms as phosphoric acid, ammonia, urea, sulphuric acid and potassium chloride.

Turkey depends on imported raw materials for its fertilizer production as the local sources are not fully explored. It is more economical for Turkey to import the raw materials and produce most fertilizers domestically than to import the fertilizer itself.

The developed world, and more particularly North America and the USSR have a dominant position in the supply of all fertilizer raw materials.

5.5.1 Phosphoric Acid

The only source of phosphorus for fertilizer production is phosphate rock. Although phosphate rock is a fairly abundant mineral in the earth's crust, the majority of world production is concentrated in four countries of the world, the USA, the USSR, Morocco and China.

Most of the Turkish phosphate deposits are difficult to mine, the rock is low in phosphorous content and also contains impurities that can be removed only by means of expensive beneficiation processes. As a result, Turkish phosphate fertilizer production is currently based on imported phosphate rock.

Toros imports phosphoric acid mainly from 3 countries, namely Morocco, South Africa and Tunisia.

5.5.2 Ammonia

While ammonia is often considered a raw material for the fertilizer industry, it is in fact an intermediate, and the key raw materials for nitrogen fertilizers are hydrocarbons. Hydrocarbons are abundant and there are no anticipated shortages of hydrocarbons for the production of nitrogenous fertilizers.

The USSR, Latin America and West Europa are the regions where ammonia production reaches to highest levels.

The ammonia requirement of Toros is met by imports, mainly from Libya, USSR and Trinidad.

5.5.3 Urea

In the fertilizer sector urea has been accepted as a primary source of nitrogen. It is the most economic nitrogen fertilizer suitable for direct application to the soil and, at the same time, of use in compound fertilizer production.

With so many urea plants around the world it is not practical to discuss individual producers. Yet, it remains true that Western European Countries are the biggest exporters of urea, with the Communist block countries, USA and Japan close behind.

Toros procures its requirements from Italy, Libya, Hungary, Romania, and USSR.

5.5.4 Sulfuric Acid

The main raw material for sulphuric acid is sulphur. Approximately 92 per cent of sulphur is used in sulphuric acid production, and of this acid production about 60 per cent is for phosphate fertilizer.

Due to its relatively small deposits and the small production of sulphur as a by-product of oil refinery operation, Turkey can meet only a minor portion of its sulphur requirements for fertilizer production.

Toros supplies its sulphuric acid requirements mainly from Algeria, Italy and Spain.

5.5.5 Potassium Chloride

Potash fertilizer production is tied to potash salt deposits which are even more localized than phosphate rock mines. In the developing countries potash is produced almost, exclusively in Europa, the USA, the USSR, Canada and Israel.

The potassium chloride requirement of Toros is met by imports, mainly from Jordan and Israel.

VI. PRODUCTION CONTROL SYSTEM FOR TOROS

Production planning is the function of management which plans, directs and controls the material supply and processing activities of an enterprise, over some future time period, in response to the forecasted demand for the products or services.

Without an accurate sales program it is difficult to achieve the optimum use of labour, plant and capital. The decisions how many to sell and when to sell them are subject first to the limitations of labour, plant and capital capacity. Inside this limitation, the extent of the total market and the proportion which can be attacked at reasonable sales expense, are factors which will affect the final decision.

If a production program is adapted which is identical with the sales program, good capital utilization will be achieved, but there will generally be considerable losses due to uneven utilization of plant and labour. If the sales requirement dates are ignored and a production program is developed which succeeds in producing the right quantities during the year, and in achieving good labor and plant utilization by producing all products at an even rate, capital utilization will be very poor, because finished products will be produced during periods when they can not be sold and large stocks will accumulate.

Production planning is the job of developing a production program which will give delivery of products at the times required to meet the sales program. Inside this primary aim, the program should be so arranged that it gives the best possible compromise between even loading of labour and plant and the optimum use of capital.

The characteristics of fertilizer production of Toros have been explained in detail in Chapter V. To summarize, there are three end products, NPK (20.20.0), NPK (15.15.15) and DAP. MAP is an intermediate product utilized in compound fertilizer production. Phosphoric acid, ammonia, urea, sulphuric acid and potassium chloride are the basic raw materials and, coating oil, coating dust, filler and polyethylene bags are the auxiliary raw materials. The production complex consists of three main units: (a) NPK Unit; (b) DAP Unit; and (c) Bagging Terminal.

The composition of the planning function such as the techniques used, and the degree of detail depends in part on the planning horizon or the length of the period for which plans are to be made. The minimum planning horizon generally covers a term ahead which is short enough to make accurate sales forecasting possible, but still long enough so that if a change of plans is necessary in the future it can be made without excessive obsolescence or an increase in capital tie-up. The planning horizon for Toros's production system has been determined as three months or nine ten-days

periods, depending on the degree to which fertilizer demand can be reasonably forecasted. This planning horizon is long enough to cover the cumulative lead time down through the product structure. The periods are termed as I/1, I/2 .. III/3.

Fertilizer sales over the planning horizon have been forecasted based on past performance, considering also the domestic and international bids that have already resulted in favour of Toros. Other relevant data such as the annual fertilizer usage rates and prices, etc. have been determined considering the Toros budget for year 1987 as a base source. (Appendix B)

6.1 ABC Analysis

Inventory analysts claim that about 80 per cent of a firm's inventory costs are caused by 20 per cent of its inventory items. The extent to which the inventory of an item is controlled should depend on the relative importance of the item on some scale. Today, with the computer available, posting of inventory records is not expensive. Nevertheless, the principle of separating the vital few is still as valid as it ever was. Aggregate inventory management requires management attention; certainly management attention ought to be devoted to those inventory categories that generate the greatest cost.

ABC Analysis involves ranking inventory items by annual costs of usage, so that the vital few items that constitute most of the inventory activity can be given the highest level of management attention, and the trivial many correspondingly less attention.

6.2 Inventory Planning

For our proposes, an inventory may be considered an accumulation of a commodity that will be used to satisfy some future demand for that commodity. Analysis of inventory problems initially involves a study of the system to determine its basic structure and boundries, the objectives thought important in the system operation, and the variables that affect the degree of attainment of these objectives.

Inventory control in Toros is concerned with the control of the quantities and monetary values of all raw materials (phosphoric acid, ammonia, urea, sulphuric acid, potassium chloride, coating oil, coating dust, filler and polyethylene bags), intermediate product (MAP) and end products (NPK (20.20.0), NPK (15.15.15) and DAP) at pre-determined levels or within safe limits.

Inventorying of a commodity should be justified by benefits accruing from one or more functions served by inventories. Among the more important reasons for inventories are those related to economies of scale in production and procurement to fluctuating requirements over time, to a desire for flexibility in scheduling sequential facilities, to speculation on price or cost, and to uncertainty about requirements and replenishment lead times.

Inventories are mostly carried because of uncertainties that exist. There are many sources of uncertainty in inventory processes. Perhaps the most common one for Toros is the demand process. Although future fertilizer consumption in Turkey and Toros's market share can be projected from past performance to some extent, degree of fertilizer sales usually are not known with certainty, but must be forecasted and these forecasts are, of course, subject to error.

Uncertainty that makes carrying inventory attractive may also result from sources related to resupply process. As mentioned previously, Toros imports all of its vital raw materials. The lead time for these items range from three to four ten-days periods with an observed variance of about a period. Therefore safety stock to avoid shortages when the lead time is unusually long may be justified.

Inventories are sometimes carried because of anticipated changes in the cost of commodities. The share of raw materials in the final cost of a fertilizer product is relatively high. Fertilizer raw material prices are very sensitive to demand/supply trends and fluctuations have been observed quite often in the past. Therefore, it is preferred to hold a certain amount of finished goods inventory to avoid raw material purchases during periods of peak prices.

In addition, when faced with a demand that fluctuates over time, as in the case of a seasonal product like fertilizer, it is desirable to build inventory during periods of low demand in anticipation of peak demand in coming periods. This build-up inventory permits Toros to operate at a fairly constant production rate over the entire season, thereby avoiding the costs of changing production rates. This inventory also helps to avoid overtime, subcontracting or lost sales to other fertilizer producers and backorder penalties during periods of peak demand.

The focal issue of inventory control is to fix, for every item, separate values for each of the following stock levels : (a) economic order quantity, (b) order point, and (c) safety stock.

(A) Economic Order Quantity

Many companies with highly successful production and inventory management systems use some order quantities set by judgement. Getting the right material to the right place at the right time can generate very significant results even though the order quantities have not been computed 'scientifically'. Getting the right quantity at the wrong time does not accomplish anything. As a consequence, economic order quantity calculations get much less attention today.

Independently calculated order quantities sometimes have a disadvantage in that they usually will not match requirements. Fertilizer production is a continuous process and compound fertilizers and DAP are being produced independently in different production units. In the compound fertilizer unit, the set-up period in changing over from NPK (20.20.0) production to NPK (15.15.15), or vice versa, takes three to four hours, which is negligibly small. Therefore, the lot-sized production method is not practiced. As for auxiliary raw materials, they are procured easily from near-by sources at virtually no fixed component of replenishment costs; and thus they are replaced only when depleted. Therefore, except in cases of imported raw materials, it is really more important to order the requirement quantity than the economic order size in Toros, and the production planning system have been computerized in this manner.

For the theoretical calculation of the economic order quantities for imported raw materials, the deterministic single item model with static demand and no backlogging case is applied, so the equations involved are :

$$EOQ = \left[\frac{2 \times RV \times UV}{i \times PV} \right]^{1/2} \quad (1)$$

where EOQ = Economic Order Quantity

RV = Fixed Component of Replenishment
Order Costs

UV = Annual Usage Rate

PV = Unit Variable Cost of Purchase,

i = Annual Inventory Carrying Cost Rate

and,

$$SV = N \times UV \times (VV)^{1/2} \quad (2)$$

where SV = Safety Stock Level

UV = Annual Usage Rate

VV = Variance of Lead Time

N = Constant (depending on the

ABC code of the inventory item).

The best particular policy structure and the best values of the parameters of the policy for a given situation will, of course, depend on the nature of the economic losses affected by the policy. Certain revenues and costs are affected by the choice of an inventory policy.

The cost of carrying inventory is high in the fertilizer industry. In Toros, the major classes of costs are inventory holding, procurement, shortage and system operating costs. Inventory holding costs include deterioration, inventory taxes, inventory insurance, storage costs and of course, cost of capital. Fortunately, the obsolescence factor is not valid for the fertilizer industry, as the required fertilizer specifications do not normally change over years. The daily cost of storage, per physical ton is higher for the high-analysis fertilizers, i.e, the products of Toros, because the greatest part of the storage cost is the interest cost based on the products value.

Procurement costs both in purchasing and production situations consist of a component that is independent of the procurement lot size and a component that varies with the lot size. The fixed components of Toros's procurements for the imported raw materials are very high. The imported raw materials are mostly delivered by sea and it should be obvious that the greatest portion of the sea transportation costs occur independently of tonnage carried.

If a demand for a temporarily out-of-stock item occurs, the economic loss depends on whether the shortage is backordered, is satisfied by substitution of a similar item or is cancelled by the customer. In all cases, there may be added information processing costs and loss of goodwill because of customer dissatisfaction. An important, but

difficult to analyze function of inventory control is to improve service to customers. Availability of fertilizer supplies in the right combination, at the right place and at the right time is extremely important. To obtain the expected performance, fertilizers should be applied to the soil at the 'right time'. Thus when a demand is not met by Toros, it is more probable that the sale will be lost to other producers whose products are more or less the same than will be backlogged.

When a series of production facilities are managed, inventories of the material in process between facilities will permit more independence in the scheduling of these facilities. Both for this reason and for the sake of attaining flexibility in converting the intermediate material, which is utilized by more than one product, to the actually demanded product, Toros prefers to hold in inventory of larger quantities of MAP, than of NPK (20.20.0) or of NPK (15.15.15).

It needs not need mentioning that inventory control also has close ties with the financial function. The stocks in a factory have all cost money and they represent a certain amount of capital tied up in production. The inventory control must be so administered that it causes the stock investment to be maintained at an optimum level.

(B) Order Point

There are two fundamental methods for scheduling, the order point system and the material requirements planning (MRP). These two methods are quite different in the way they handle ordering. The order point system assumes that each item in inventory is independent of all other items and can be ordered independently. MRP treats items as if their priorities are dependent and it calculates future demands. The 'independent/dependent demand' principle states that the order point method should be used only on independent demand items, while MRP should be used for dependent demand items.

The time from the moment it is determined that an item needs to be replenished until the time that it is determined that is back in inventory and once more available is termed as the 'lead time'. The order point method is based on some forecast of demand over the replenishment lead time and adding safety stock to it. When the total stock on hand plus on order falls below the order point, a new supply is ordered. Time phased MRP shows planned orders in their proper release times based on the lead time. A planned order is created when the projected available balance goes below zero or safety stock. In other words, planned orders are offset by lead time from the net requirements.

When a reorder is placed this time depends on the quantity of the order placed last time. Time phasing the order point puts the emphasis on keeping the priority correct on open orders, facilitates capacity and lower level materials planning, and handles lumpy known future demands.

(C) Safety Stock

To protect against losses, resulting from shortages when the actual demand exceeds the forecast, it is often economic to carry inventory above the level that would result from planning. This extra inventory is called the safety stock, and the amount required is a function of the uncertainty in the forecast.

It is possible to use the safety stock technique with MRP. It is important to remember, however, that safety stocks tend to dilute priorities. With the designed MRP system, much less safety stock is needed as it can reasonably be calculated over the shortest reasonable lead time. When lead time can be varied, it becomes more important to plan ahead and replan priorities than to compute safety stocks precisely. In cases when actual demand is less than the forecast, safety stock represents unnecessary additional inventory investment. MRP reacts to increases and decreases in demand by revising need dates on open replenishment orders.

In constructing the MRP system for Toros, safety stocks are kept to a minimum and are planned only in the master schedule. Additional safety stocks for the low level items are not usually planned, since they have already been included in the master schedule, in terms of the final fertilizer products. In the model it is assumed that there is no storage requirement if fertilizer is produced and used during the same period. Thus, safety stock levels for only NPK (20.20.0), NPK (15.15.15), DAP and MAP are determined to meet ten days requirements.

Another fact which is considered in designing the inventory control system for Toros is the maximum storage capacities. Raw materials for fertilizer production require specially designed storage facilities. The existing complex includes storage tanks for phosphoric acid, ammonia and sulphuric acid. The planned orders for these items are released considering the maximum available storage capacities at that time.

6.3 Bill of Materials

Material Requirements Planning requires a Bill of Materials (BOM). BOM identifies how the end product is manufactured, specifying all subcomponents, their sequence of build-up and their quantity in each finished unit. BOM is a structured parts list that shows what material 'goes into' the product and represents the way the product flows. The output of the Bill of Materials program is a lower triangular matrix whose elements show the total requirements of each inventory item. The procedure of the Bill of Materials program is explained in detail in Segem's study.(8)

6.4 Material Requirements Planning

Scheduling is the key function in production and inventory management. The function of scheduling is to specify when, in calendar time, certain events are to take place with respect to both the inputs and outputs of a system. The decisions about the timing and sequencing of work must be related to certain measures of system performance. A workable, formal system can be used to respond better and faster to the needs of the marketplace, to reduce inventory, and to improve productivity.

Timing is particularly important where priorities are dependent, where there are parts that go into assemblies, where a semifinished component must be available in order to manufacture the finished component.

Today with computers readily available, and low cost file storage, to computerize the MRP system is a reasonable approach. The computer using MRP takes a master schedule, looks up the bill of material in its own files to find out what material is required to manufacture the product in the schedule, and then checks the inventory to see if the material is on hand. If the material is not on hand, it tells the planner to order it and tells him when it will be needed. If the material is already on order; it will

reevaluate the due dates on that material and tell the planner if the due dates need to be changed.

Probably the most important aspect of MRP is the master schedule. For out in the future the master schedule is usually a forecast of anticipated production. Over the near term, it is a fairly precise statement of what the company intends to produce; since actual customer orders were used as the basis for the master schedule. As actual customer orders do come in, actual known demand replaces forecast as the prime reference in making a master schedule. The master schedule is not a one time statement of some production objectives, it is an ever changing handle that enables us to manage a modern production and inventory management system. Master scheduling is the job of fixing starting and finishing times for the work on the major product units. The master schedule produced at this stage is in effect a timetable, which shows when the work on each major part of the product must be started and finished if the whole contract is to be completed by the required delivery or completion date. When a planned order moves into the current period and is released, it is converted to a 'scheduled receipt'.

The MRP program references the bill of materials in the master schedule and thus material requirements are generated. In a time-phased MRP system, gross requirements are netted, that is deducted from the inventory on hand and

on order scheduled receipts to project an available balance. When requirements exceed the amount on hand and on order, a negative balance, or net requirement results. The net requirement is converted to a lot size if necessary and is then offset by the lead time to create a planned order release. Planned order releases at one level in the product structure generate material requirements in the proper time periods at lower levels. The planned order releases from MRP are used as input to the CRP program in return.

The master schedule itself should be revised if the required material or capacity is not available. Modern MRP is an approach for calculating material requirements not only to generate replenishment orders, but also to reschedule open orders to meet changing requirements. Thus when using MRP, uncertainty exist only at the master schedule level. Schedules for all supporting material will be geared to the master schedule. Once a program has been adapted, the next job is to place orders on the buyer and on the processing departments for the materials and parts needed to make the products, and to so arrange the ordering quantities and delivery schedules that all items are delivered in time to meet the production program.

6.5 Capacity Requirements Planning

The task of CRP is to match the production plans and schedules with the productive capability of the production centers. The capacity of a production center is determined by the manpower and facilities comprising that center. At a production center, operations are performed that utilize a certain amount of the capacity of the center for each unit of product processed.

A time phased MRP not only releases orders, but it also generates planned orders that are used to create lowerlevel material requirements. CRP can be made then by taking into account the hours by production center by time period needed to produce the planned order releases. CRP makes a tentative plan to show the capacity that is needed. This can be compared with the actual capacity available to determine whether or not the master schedule can be met. In the event the required amount of capacity is not available, the detailed CRP report showing the individual released orders making up the capacity requirements for a given time period will be referenced. The results of the program are useful for the decisions on increasing the number or shifts, working overtime, or altering the master plan. The master schedule can then be revised and a new MRP and CRP program can be run.

The CRP program has been so designed that the capacities of the Toros production units are measured in terms of inputs of the system, such as machine or labour hours. At any given time, the theoretical capacities of the Toros production units, that is the standards of the fertilizer production process, are adjusted to a normal or net operating capacity. The latter allows for operating interruptions such as absence, vacations, ordinary repairs, set-ups, machine failures, material delivery delays and management errors.

The production activities in the Toros NPK and DAP units are continuous for 24 hours, while the number of shifts worked in the bagging terminal varies depending on the work load. To aid the following steps in decision making, the CRP module is programmed to show the extent of operating time in excess of nominal capacity (one shift/day) and in excess of maximum capacity (three shifts/day) or further available time of the production unit.

VII. SUMMARY OF FINDINGS

The results of the production planning system, specifically designed for Toros, are submitted in Appendix C as computer outputs in tabular form. The conclusions derived on each module of the production planning system are summarized in the following sections.

7.1 ABC Analysis

As previously explained in more detail, ABC Analysis groups inventory items into three categories, and determines the degree of control for each category. The results of the ABC Analysis for the fertilizer manufacturing company can be summarized in the following table :

'A' ITEMS	'B' ITEMS	'C' ITEMS
NPK (20.20.0)	NPK (15.15.15)	Ammonia
MAP	DAP	Sulphuric Acid
Phosphoric Acid		Potassium Chloride
Urea		Coating Oil
		Coating Dust
		Filler Material
		Polyethylene Bags

The ABC Analysis reveals that NPK (20.20.0), MAP, phosphoric acid and urea, constitute most of the inventory activity and thus are categorized as 'A' items. Despite the common belief that the share of raw materials in the total cost of final fertilizer product is high, the ABC Analysis shows that the total raw material cost is mostly made up of two items, phosphoric acid and urea. The inventory control system for the 'A' items should have the following

characteristics : (a) small safety stock level; (b) frequent inventory review; (c) frequent orders; (d) detailed records; and (e) most capable personnel.

The results of the ABC Analysis also reveal that the other two products of Toros, namely compound fertilizers NPK (15.15.15) and DAP should be treated as 'B' items. All the other raw materials of fertilizer production, most of which are required in small quantities and are fairly abundant, are grouped under 'C' category. Characteristics of 'C' items will be the opposite of 'A' items and characteristics of 'B' items will fall between those of 'A' and 'C' items, that is : (a) medium quantity of safety stock; (b) less frequent review; (c) less frequent orders; (d) less detailed records; and (e) average personnel.

7.2 Inventory Planning

As it is known, the focal issue of inventory control systems is to determine the economic order quantity, order point and safety stock levels for the inventory items. It is obvious that the production control system designed specifically for a company should consider the basic structure and constraints. As explained in detail in the previous sections, economic order quantities are calculated only for the imported raw materials in Toros. Other raw materials are procured easily from near-by sources and are replenished as required. In addition, safety stocks are kept only for the intermediate and end products and at a minimum level, being dependent on the ABC category of the item. It is interesting to note that the findings of the designed inventory control system coincides with the common practice in Toros. The outputs of the Inventory Planning, i.e. the economic order quantity, lead time and safety stock level of each inventory item are inputted to the MRP program.

7.3 Bill of Materials

The computer output of the Bill of Material programming, submitted in Appendix D is self-explanatory and requires no further comments. Indeed the outputs of this program do not have a significant meaning by itself, but they are used as data for the MRP program module.

7.4 Material Requirements Planning

The material requirements planning computer program has three time-phased output tables; namely, Net Requirement Amounts, Gross Requirement Amounts, and finally, Material Requirements Planning. The values for each item in the Net Requirement Amounts table provide the answer to the simple question such as 'how much of urea is needed to produce 100 tons each of NPK (20.20.0), NPK (15.15.15) and DAP?'. In the calculation of these values, only the build-up sequence and process quantities are considered. The Gross Requirement Amounts table is an intermediate step in the Material Requirements Planning. As the planned orders are released at one level in the product structure, gross requirements are generated at lower levels in the proper time units; and the final requirements for these low value items can be observed from the Gross Requirement Amounts table. The Material Requirements Planning table goes one step further and releases orders considering the economic order quantities of the inventory items. The production and procurement orders should be given in agreement with this output.

7.5 Capacity Requirements Planning

The CRP program matches the MRP results with the productive capability of the production units. When the time-phased capacity requirements planning output table is observed, it is noticed that the work load of the NPK and MAP production units and that of the bagging terminal exceeds the maximum man-hour and machine-hour capacity of 75 hours made up by three shifts per day. On the other hand, production time is further available at the DAP unit.

7.4.1 NPK and DAP Units

The work load of the NPK unit, that is in excess of the maximum available capacity has been calculated in the program as :

Hours Exceeding Maximum Capacity :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
0	0	87	111	159	159	0	39	39

Converted into product tons, using the average processing times, the excess work in the NPK unit becomes :

Tons Exceeding Maximum capacity :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
0	0	3625	4625	6625	6625	0	1625	1625

As the DAP unit has the flexibility of producing compound fertilizers, it is possible to transfer the excess production load of the NPK unit to the DAP unit. The compound fertilizer production rate in both units is equal; i.e. 0.024 mtpd. The set-up time required to shift from DAP production to NPK production is approximately 4 hours. Thus, as the work load of the NPK unit is reduced to the level of maximum capacity, that of the DAP unit becomes :

Required Operating Time (DAP production) :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
21	40	40	40	40	40	40	40	40

Required Operating Time (NPK production) :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
0	0	87	111	159	159	0	39	39

Setup Time

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
0	0	4	4	4	4	0	4	4

Thus, total operating time required in the DAP unit becomes :

Total Required Operating Time :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
21	40	131	155	203	203	40	83	83

Time Exceeding Nominal Capacity :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
0	0	48	80	128	128	0	80	80

Time Exceeding Maximum Capacity :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
0	0	0	0	0	0	0	0	0

Time Further Available :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
54	35	0	0	0	0	35	0	0

As can be observed from the last two calculations, as the work load of the NPK production unit is reduced to the level of maximum capacity, the productive capability of the DAP unit has been further utilized. Similar calculations can be tried on the results of the CRP program to finalize the production plan. In cases when it is not possible to match the production plans with the productive capability of the production centers, the master schedule should be revised; and the MRP program should be re-run until more satisfactory results are obtained.

7.4.2 MAP Unit

The work load of the MAP unit, that is in excess of the maximum available capacity, has been calculated in the program as :

Hours Exceeding Maximum Capacity :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
0	10	35	74	74	0	0	0	0

Converted into product tons, using the average processing times, the excess work in the MAP unit becomes :

Tons Exceeding Maximum capacity :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
0	200	690	1475	1475	0	0	0	0

It is suggested that instead of revising the master schedule,

as the available MAP

production capacity within the Toros premises is not sufficient to meet the existing demand during some periods of the planning horizon, the above calculated amounts of MAP should be procured from external sources.

7.4.3 Bagging Terminal

As can be better observed from the program output, the existing production plans do not fully utilize the capacity of the bagging terminal.

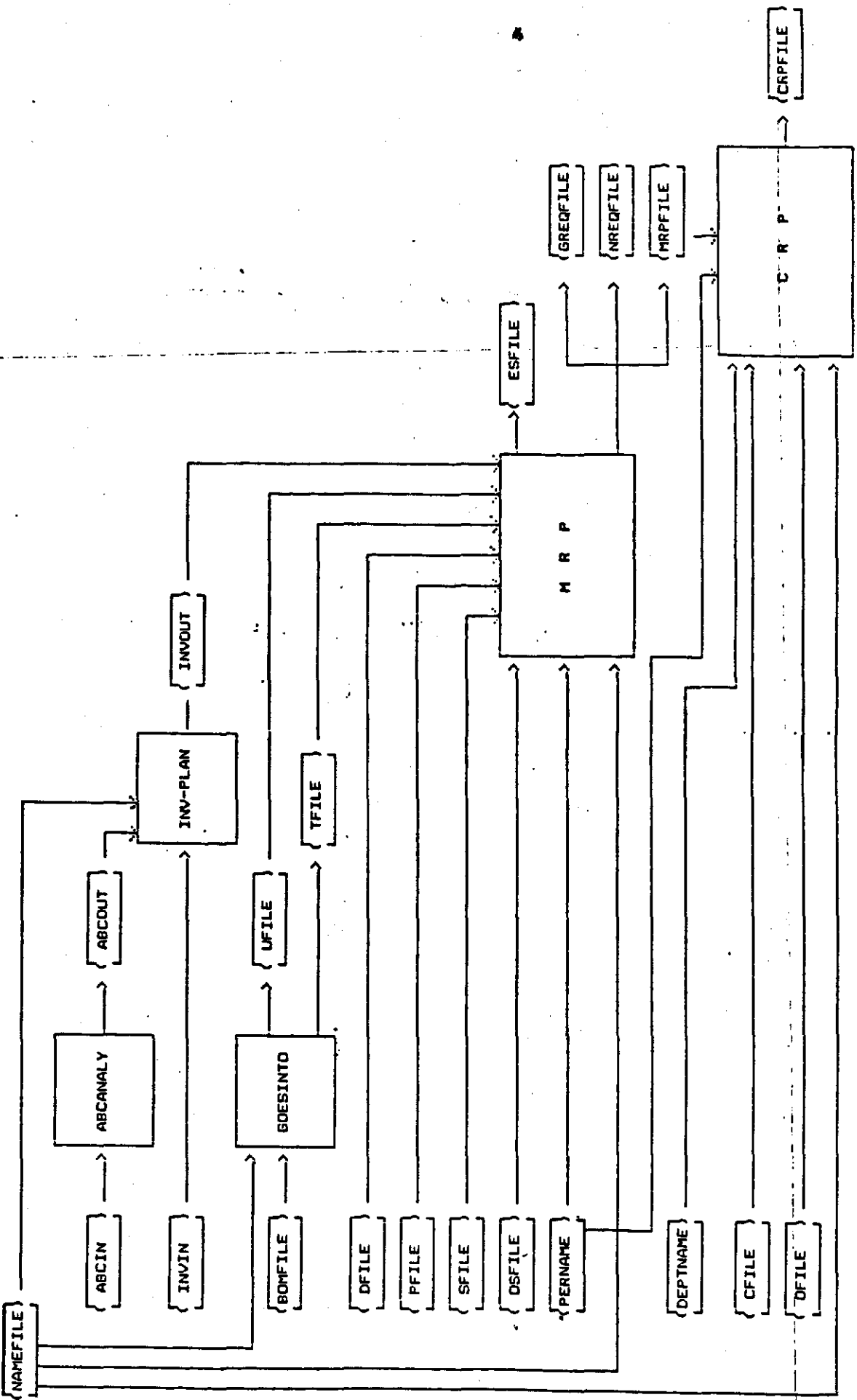
Time Further Available :

<u>I/1</u>	<u>I/2</u>	<u>I/3</u>	<u>II/1</u>	<u>II/2</u>	<u>II/3</u>	<u>III/1</u>	<u>III/2</u>	<u>III/3</u>
67	39	19	15	7	7	35	27	27

Thus, possibilities of utilizing the excess time available in the bagging terminal by providing service to third parties should be sought.

A P P E N D I X A

FILE NETWORK DIAGRAM



NAME : RECORD

REC_NO : INTEGER

CODE : STRING4

NAME : STRING20

UNIT : STRING4

CLASS : STRING4

DEPTNAME : RECORD

REC_NO : INTEGER

DEPTNAME : STRING20

PERNAME : RECORD

REC_NO : INTEGER

PERNAME : STRING20

ABCIN : RECORD

REC_NO : INTEGER

CODE : STRING4

PRICE : REAL

USAGE : REAL

ABCOUT : RECORD

CODE : STRING4

PRICE : REAL

USAGE : REAL

ABCCODE : CHAR

INVIN : RECORD

REC_NO : INTEGER

CODE : STRING4

REPLCOST : REAL

LEADTIME : INTEGER

LEADVARIANCE : REAL

INVOUT : RECORD

CODE : STRING4

ORDQUANT : REAL

LEADTIME : INTEGER

SSTOCK : REAL

BOMFILE : RECORD

REC_NO : INTEGER

FATHERCODE : STRING4

SONCODE : STRING4

GOESINTO : REAL

UFILE : REAL

TFILE : REAL

D F I L E : R E A L

P F I L E : R E A L

S F I L E : R E A L

O S F I L E : R E A L

E S F I L E : R E A L

GREQFILE : REAL

NREQFILE : REAL

MRPFILE : REAL

C F I L E : R E A L

O F I L E : R E A L

C R P F I L E : R E A L

A P P E N D I X B

D A T A F I L E S

NO	CODE	NAME	UNIT	CLASS
---	---	---	---	---
1	001	20.20.0	TON	M
2	002	15.15.15	TON	M
3	003	DAP	TON	M
4	004	NAP	TON	M
5	005	PHOSPHORIC ACID	TON	P
6	006	AMMONIA	TON	P
7	007	UREA	TON	P
8	008	SULPHURIC ACID	TON	P
9	009	POTASSIUM CHLORIDE	TON	P
10	010	COATING OIL	TON	P
11	011	COATING DUST	TON	P
12	012	FILLER MATERIAL	TON	P
13	013	POLYETHYLENE BAGS	TON	P

NAME File

Number of Records : 13

<u>NO</u>	<u>PERIOD NAME</u>
1	I/1
2	I/2
3	I/3
4	II/1
5	II/2
6	II/3
7	III/1
8	III/2
9	III/3

PERNAME File

Number of Records : 9

<u>NO</u>	<u>PRODUCTION UNIT</u>
1	MPK UNIT
2	MAP UNIT
3	DAP UNIT
4	BAGGING TERMINAL

DEPTNAME File Number of Records : 4

<u>NO</u>	<u>NAME</u>	<u>CODE</u>	<u>PURCHASE PRICE</u>	<u>ANNUAL USAGE</u>
1	20.20.0	001	144000.00	416000.00
2	15.15.15	002	133200.00	70000.00
3	DAP	003	192000.00	44000.00
4	MAP	004	175000.00	100000.00
5	PHOSPHORIC ACID	005	270000.00	117000.00
6	AMMONIA	006	99000.00	53000.00
7	UREA	007	81000.00	129000.00
8	SULPHURIC ACID	008	45000.00	13000.00
9	POTASSIUM CHLORIDE	009	76500.00	20000.00
10	COATING OIL	010	156000.00	840.00
11	COATING DUST	011	52000.00	4200.00
12	FILLER MATERIAL	012	1000.00	121700.00
13	POLYETHYLENE BAGS	013	1085000.00	2055.00

ABCIN File

Number of Records :13

<u>NO</u>	<u>CODE</u>	<u>REPLENISHMENT COST</u>	<u>LEAD TIME</u>	<u>LEAD VARIANCE</u>
1	001	0.000	0	1.00
2	002	0.000	0	1.00
3	003	0.000	0	1.00
4	004	0.000	1	1.00
5	005	50000000.000	3	0.00
6	006	50000000.000	3	0.00
7	007	10000000.000	3	0.00
8	008	10000000.000	3	0.00
9	009	10000000.000	3	0.00
10	010	0.000	0	0.00
11	011	0.000	0	0.00
12	012	0.000	0	0.00
13	013	0.000	0	0.00

INVIN File

Number of Records : 13

NO	FATHERCODE	SONCODE	GOESINTO
--	-----	-----	-----
1	001	004	0.392
2	001	006	0.044
3	001	007	0.271
4	001	008	0.055
5	001	010	0.003
6	001	011	0.015
7	001	012	0.250
8	001	013	0.004
9	002	004	0.293
10	002	006	0.017
11	002	007	0.231
12	002	008	0.032
13	002	009	0.252
14	002	010	0.003
15	002	011	0.015
16	002	012	0.178
17	002	013	0.004
18	003	005	0.481
19	003	006	0.228
20	003	010	0.003
21	003	011	0.015
22	003	012	0.060
23	003	013	0.004
24	004	005	0.572
25	004	006	0.155
26	004	010	0.003
27	004	011	0.015

CODE	NAME	STORAGE CAPACITY
----	----	-----
001	20.20.0	0.000 TON
002	15.15.15	0.000 TON
003	DAP	0.000 TON
004	MAP	0.000 TON
005	PHOSPHORIC ACID	0.000 TON
006	AMMONIA	0.000 TON
007	UREA	0.000 TON
008	SULPHURIC ACID	0.000 TON
009	POTASSIUM CHLORIDE	0.000 TON
010	COATING OIL	0.000 TON
011	COATING DUST	0.000 TON
012	FILLER MATERIAL	0.000 TON
013	POLYETHYLENE BAGS	0.000 TON

<u>CODE</u>	<u>NAME</u>	<u>ON HAND STOCK</u>
001	20.20.0	15783.000 TON
002	15.15.15	4505.000 TON
003	DAP	1684.000 TON
004	MAP	1900.000 TON
005	PHOSPHORIC ACID	11783.000 TON
006	AMMONIA	5011.000 TON
007	UREA	15480.000 TON
008	SULPHURIC ACID	2380.000 TON
009	POTASSIUM CHLORIDE	3000.000 TON
010	COATING OIL	23.000 TON
011	COATING DUST	105.000 TON
012	FILLER MATERIAL	900.000 TON
013	POLYETHYLENE BAGS	125.000 TON

OPERATION TIMES AT PRODUCTION UNITS

(HOURS)

NAME	NPK UNIT	MAP UNIT	DAP UNIT	BAGGING TERMINAL
20.20.0	0.024	0.000	0.000	0.004
15.15.15	0.024	0.000	0.000	0.004
DAP	0.000	0.000	0.040	0.004
MAP	0.000	0.050	0.000	0.000
PHOSPHORIC ACID	0.000	0.000	0.000	0.000
AMMONIA	0.000	0.000	0.000	0.000
UREA	0.000	0.000	0.000	0.000
SULPHURIC ACID	0.000	0.000	0.000	0.000
POTASSIUM CHLORIDE	0.000	0.000	0.000	0.000
COATING OIL	0.000	0.000	0.000	0.000
COATING DUST	0.000	0.000	0.000	0.000
FILLER MATERIAL	0.000	0.000	0.000	0.000
POLYETHYLENE BAGS	0.000	0.000	0.000	0.000

A P P E N D I X C

O U T P U T F I L E S

ABC ANALYSIS FOR INVENTORY ITEMS

NO	CODE	NAME	UNIT	CLASS	PURCHASE PRICE	ANNUAL USAGE	ABCCODE
1	001	20.20.0	TON	H	144000.00	416000.00	A
2	002	15.15.15	TON	H	133200.00	70000.00	B
3	003	DAP	TON	H	192000.00	44000.00	B
4	004	MAP	TON	H	175000.00	100000.00	A
5	005	PHOSPHORIC ACID	TON	P	270000.00	117000.00	A
6	006	AMMONIA	TON	P	99000.00	53000.00	C
7	007	UREA	TON	P	81000.00	129000.00	A
8	008	SULPHURIC ACID	TON	P	45000.00	13000.00	C
9	009	POTASSIUM CHLORIDE	TON	P	76500.00	20000.00	C
10	010	COATING OIL	TON	P	156000.00	840.00	C
11	011	COATING DUST	TON	P	52000.00	4200.00	C
12	012	FILLER MATERIAL	TON	P	1000.00	121700.00	C
13	013	POLYETHYLENE BAGS	TON	P	1085000.00	2055.00	C

INVENTORY PLANNING

<u>NO</u>	<u>CODE</u>	<u>NAME</u>	<u>UNIT</u>	<u>CLASS</u>	<u>ORDER QUANTITY</u>	<u>LEAD TIME</u>	<u>SAFETY STOCK</u>
1	001	20.20.0	TON	M	0.000	2	5777.778
2	002	15.15.15	TON	M	0.000	2	1944.444
3	003	DAP	TON	M	0.000	2	1222.222
4	004	MAP	TON	M	0.000	2	1388.889
5	005	PHOSPHORIC ACID	TON	P	8164.966	3	0.000
6	006	AMMONIA	TON	P	9075.356	3	0.000
7	007	UREA	TON	P	7000.203	3	0.000
8	008	SULPHURIC ACID	TON	P	2981.424	3	0.000
9	009	POTASSIUM CHLORIDE	TON	P	2836.238	3	0.000
10	010	COATING OIL	TON	P	0.000	0	0.000
11	011	COATING DUST	TON	P	0.000	0	0.000
12	012	FILLER MATERIAL	TON	P	0.000	0	0.000
13	013	POLYETHYLENE BAGS	TON	P	0.000	0	0.000

INVENTORY PLANNING

<u>NO</u>	<u>CODE</u>	<u>NAME</u>	<u>UNIT</u>	<u>CLASS</u>	<u>ORDER QUANTITY</u>	<u>LEAD TIME</u>	<u>SAFETY STOCK</u>
1	001	20.20.0	TON	M	0.000	0	5800.000
2	002	15.15.15	TON	M	0.000	0	2000.000
3	003	DAP	TON	M	0.000	0	1200.000
4	004	MAP	TON	M	0.000	1	1400.000
5	005	PHOSPHORIC ACID	TON	P	8000.000	3	0.000
6	006	AMMONIA	TON	P	9000.000	3	0.000
7	007	UREA	TON	P	7000.000	3	0.000
8	008	SULPHURIC ACID	TON	P	3000.000	3	0.000
9	009	POTASSIUM CHLORIDE	TON	P	3000.000	3	0.000
10	010	COATING OIL	TON	P	0.000	0	0.000
11	011	COATING DUST	TON	P	0.000	0	0.000
12	012	FILLER MATERIAL	TON	P	0.000	0	0.000
13	013	POLYETHYLENE BAGS	TON	P	0.000	0	0.000

NET REQUIREMENT AMOUNTS

NO	CODE	NAME	UNIT	CLASS	NET AMOUNTS REQUIRED								
					I/1	I/2	I/3	II/1	II/2	II/3	III/1	III/2	III/3
1	001	20.20.0	TON	M	7000.000	7000.000	9000.000	11000.000	13000.000	13000.000	6000.000	7000.000	7000.000
2	002	15.15.15	TON	M	4000.000	4000.000	4000.000	3000.000	3000.000	3000.000	3000.000	4000.000	4000.000
3	003	DAP	TON	M	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000
4	004	NAP	TON	M	3916.000	3916.000	4700.000	5191.000	5975.000	5975.000	3231.000	3916.000	3916.000
5	005	PHOSPHORIC ACID	TON	P	2722.718	2722.718	3171.480	3452.528	3901.290	3901.290	2330.624	2722.718	2722.718
6	006	AMMONIA	TON	P	1210.580	1210.580	1420.100	1567.205	1776.725	1776.725	1043.405	1210.580	1210.580
7	007	UREA	TON	P	2821.000	2821.000	3363.000	3674.000	4216.000	4216.000	2319.000	2821.000	2821.000
8	008	SULPHURIC ACID	TON	P	513.000	513.000	623.000	701.000	811.000	811.000	426.000	513.000	513.000
9	009	POTASSIUM CHLORIDE	TON	P	1008.000	1008.000	1008.000	756.000	756.000	756.000	756.000	1008.000	1008.000
10	010	COATING OIL	TON	P	47.748	47.748	56.100	60.573	68.925	68.925	39.693	47.748	47.748
11	011	COATING DUST	TON	P	238.740	238.740	280.500	302.865	344.625	344.625	198.465	238.740	238.740
12	012	FILLER MATERIAL	TON	P	2522.000	2522.000	3022.000	3344.000	3844.000	3844.000	2094.000	2522.000	2522.000
13	013	POLYETHYLENE BAGS	TON	P	45.600	45.600	53.200	57.000	64.600	64.600	38.000	45.600	45.600

GROSS REQUIREMENT AMOUNTS

NO	CODE	NAME	UNIT	CLASS	GROSS AMOUNTS REQUIRED								
					I/1	I/2	I/3	II/1	II/2	II/3	III/1	III/2	III/3
1	001	20.20.0	TON	H	7000.000	7000.000	9000.000	11000.000	13000.000	13000.000	6000.000	7000.000	7000.000
2	002	15.15.15	TON	H	4000.000	4000.000	4000.000	3000.000	3000.000	3000.000	3000.000	4000.000	4000.000
3	003	DAP	TON	H	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000
4	004	MAP	TON	H	438.035	2746.664	4700.000	5191.000	5975.000	5975.000	3231.000	3916.000	3916.000
5	005	PHOSPHORIC ACID	TON	P	1785.021	3171.480	3452.520	3901.290	3901.290	2330.624	2722.710	2722.710	481.200
6	006	AMMONIA	TON	P	550.985	1200.840	1496.205	1688.725	1776.725	1351.405	1149.580	1210.580	603.600
7	007	UREA	TON	P	345.345	2012.607	3363.000	3674.000	4216.000	4216.000	2319.000	2821.000	2821.000
8	008	SULPHURIC ACID	TON	P	47.840	348.935	623.000	701.000	811.000	811.000	426.000	513.000	513.000
9	009	POTASSIUM CHLORIDE	TON	P	376.740	1008.000	1008.000	756.000	756.000	756.000	756.000	1008.000	1008.000
10	010	COATING OIL	TON	P	14.087	41.151	57.573	62.925	60.925	60.693	41.748	47.748	36.000
11	011	COATING DUST	TON	P	70.435	205.755	287.865	314.625	344.625	303.465	208.740	238.740	180.000
12	012	FILLER MATERIAL	TON	P	297.070	1776.250	3022.000	3344.000	3044.000	3044.000	2094.000	2522.000	2522.000
13	013	POLYETHYLENE BAGS	TON	P	7.642	34.265	53.200	57.000	64.600	64.600	38.000	45.600	45.600

MATERIAL REQUIREMENTS PLANNING

NO	CODE	NAME	UNIT	CLASS	AMOUNT TO BE PRODUCED/ORDERED									AMOUNT IN STOCK
					I/1	I/2	I/3	II/1	II/2	II/3	III/1	III/2	III/3	
1	001	20.20.0	TON	N	0.000	4017.000	9000.000	11000.000	13000.000	13000.000	6000.000	7000.000	7000.000	0.000
2	002	15.15.15	TON	N	1495.000	4000.000	4000.000	3000.000	3000.000	3000.000	3000.000	4000.000	4000.000	0.000
3	003	BAP	TON	N	516.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	0.000
4	004	RAP	TON	N	2684.699	4700.000	5191.000	5975.000	5975.000	3231.000	3916.000	3916.000	0.000	0.000
5	005	PHOSPHORIC ACID	TON	P	8000.000	0.000	0.000	8000.000	0.000	0.000	0.000	0.000	0.000	3241.959
6	006	AMMONIA	TON	P	0.000	9000.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	5538.785
7	007	UREA	TON	P	0.000	0.000	7000.000	0.000	7000.000	0.000	0.000	0.000	0.000	6050.000
8	008	SULPHURIC ACID	TON	P	0.000	3000.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	982.000
9	009	POTASSIUM CHLORIDE	TON	P	3000.000	0.000	0.000	0.000	3000.000	0.000	0.000	0.000	0.000	2952.000
10	010	COATING OIL	TON	P	0.000	32.238	57.573	62.925	68.925	60.693	41.748	47.748	36.000	0.000
11	011	COATING DUST	TON	P	0.000	171.190	287.865	314.625	344.625	303.465	208.740	238.740	180.000	0.000
12	012	FILLER MATERIAL	TON	P	0.000	1173.320	3022.000	3344.000	3844.000	3844.000	2094.000	2522.000	2522.000	0.000
13	013	POLYETHYLENE BAGS	TON	P	0.000	0.000	0.000	27.106	64.600	64.600	38.000	45.600	45.600	0.000

CAPACITY REQUIREMENTS PLANNING

DEPT : WPX UNIT

	1/1	1/2	1/3	11/1	11/2	11/3	111/1	111/2	111/3
REQUIRED OPERATING TIME	35.880	192.408	312.000	336.000	384.000	384.000	216.000	264.000	264.000
TIME EXCEEDING NOM. CAPACITY (1)	0.000	117.408	237.000	261.000	309.000	309.000	141.000	189.000	189.000
TIME EXCEEDING MAX. CAPACITY (3)	0.000	0.000	87.000	111.000	159.000	159.000	0.000	39.000	39.000
TIME FURTHER AVAILABLE	39.120	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

DEPT : MAP UNIT

	1/1	1/2	1/3	11/1	11/2	11/3	111/1	111/2	111/3
REQUIRED OPERATING TIME	134.235	235.000	259.550	298.750	298.750	161.550	195.800	195.800	0.000
TIME EXCEEDING NOM. CAPACITY (1)	59.235	160.000	184.550	223.750	223.750	86.550	120.800	120.800	0.000
TIME EXCEEDING MAX. CAPACITY (3)	0.000	10.000	34.550	73.750	73.750	0.000	0.000	0.000	0.000
TIME FURTHER AVAILABLE	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	75.000

DEPT : BAP UNIT

	1/1	1/2	1/3	11/1	11/2	11/3	111/1	111/2	111/3
REQUIRED OPERATING TIME	20.640	40.000	40.000	40.000	40.000	40.000	40.000	40.000	40.000
TIME EXCEEDING NOM. CAPACITY (1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TIME EXCEEDING MAX. CAPACITY (3)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TIME FURTHER AVAILABLE	54.360	35.000	35.000	35.000	35.000	35.000	35.000	35.000	35.000

DEPT : BAGGING TERMINAL

	1/1	1/2	1/3	11/1	11/2	11/3	111/1	111/2	111/3
REQUIRED OPERATING TIME	8.044	36.068	56.000	60.000	68.000	68.000	40.000	48.000	48.000
TIME EXCEEDING NOM. CAPACITY (1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TIME EXCEEDING MAX. CAPACITY (3)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TIME FURTHER AVAILABLE	66.956	38.932	19.000	15.000	7.000	7.000	35.000	27.000	27.000

APPENDIX D

SOURCE FILES


```

Line 1 Col 1 Insert Indent A:NAME.PAS
PROGRAM NAMEDATA;
{$V-}
TYPE
  STRING4 = STRING[4];
  STRING20 = STRING[20];
  INPUTRECORD = RECORD
    REC_NO : INTEGER;
    CODE : STRING4;
    NAME : STRING20;
    UNIT : STRING4;
    CLASS : STRING4;
    NAMELENGTH : INTEGER;
  END;
  RECFILE = FILE OF INPUTRECORD;
VAR
  RECNO : INTEGER;
  CH : CHAR;
  BRANCH : CHAR;
  NL : INTEGER;
  QUIT : BOOLEAN;
  INREC : INPUTRECORD;
  NAMEIN : FILE OF INPUTRECORD;
CONST
  COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];

```

```

Line 47 Col 1 Insert Indent A:NAME.PAS

```

```

FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;
TYPE REG_PACK = RECORD
  AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
END;
VAR REGISTERS : REG_PACK;
BEGIN
  REGISTERS.AX := $0600;
  REGISTERS.DX := 255;
  MSDOS(REGISTERS);
  CH :=CHR(REGISTERS.AX);
  KEYSTAT := CH <> CHR(0)
END;

PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;

```

```

Line 70 Col 1 Insert Indent A:NAME.PAS
PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');
  GOTOXY (12,2); WRITELN ('! CTRL-C : CLEAR !');
  GOTOXY (12,3); WRITELN ('! CTRL-A : APPEND !');
  GOTOXY (12,4); WRITELN ('! CTRL-E : EDIT !');
  GOTOXY (12,5); WRITELN ('! CTRL-Q : QUIT !');
  GOTOXY (12,6); WRITELN ('! CTRL-L : LIST !');
  GOTOXY (12,7); WRITELN ('! CTRL-P : PRINT !');

```

```
END;
```

```

PROCEDURE SCREEN_APPEND;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');
  GOTOXY (12,2); WRITELN ('! F1 : SAVE !');
  GOTOXY (12,3); WRITELN ('! ESC : ESCAPE !');
  GOTOXY (12,4); WRITELN ('! CTRL-Q : QUIT !');
  GOTOXY (5,10);
  WRITELN('REC_NO :');WRITELN;
  WRITELN('CODE :');
  WRITELN('NAME :');

```

```
Line 93 Col 1 Insert Indent A:NAME.PAS
```

```

WRITELN('UNIT :');
WRITELN('CLASS :');

```

```
END;
```

```

PROCEDURE PROC_EDIT (VAR NAMEIN : RECFILE;
                     VAR INREC : INPUTRECORD;
                     VAR BRANCH : CHAR);

```

```
LABEL 1,11,21,31,41,27;
```

```
TYPE
```

```

INDEX = 1..20;
STRING4 = STRING[4];
RECNOVECTOR = ARRAY [INDEX] OF INTEGER;
CODEVECTOR = ARRAY [INDEX] OF STRING4;
NAMEVECTOR = ARRAY [INDEX] OF STRING20;
CLASSVECTOR = ARRAY [INDEX] OF STRING4;
UNITVECTOR = ARRAY [INDEX] OF STRING4;
LENGTHVECTOR = ARRAY [INDEX] OF INTEGER;

```

```
VAR
```

```

RV : RECNOVECTOR;
CV : CODEVECTOR;
NV : NAMEVECTOR;

```

Line 116 Col 1 Insert Indent A:NAME.PAS

```
KV : CLASSVECTOR;
BV : UNITVECTOR;
LV : LENGTHVECTOR;
CH : CHAR;
C  : STRING4;
ERROR : INTEGER;
EXIT : BOOLEAN;
```

CONST

```
COMMANDSET : SET OF CHAR = [#17,#27,#59];
```

BEGIN

CASE BRANCH OF

'E' :BEGIN

```
EXIT := FALSE;
CLRSCR;
GOTOXY(10,10);
WRITELN('FIND RECNO = ');
11 : GOTOXY (23,10);CLR_EOL;
WHILE NOT KEYSTAT (CH) DO BEGIN END;
IF ((ORD(CH)=27) OR (ORD(CH)=17))
    THEN GOTO 21
    ELSE BEGIN
        WRITE (CH);
        READ(C);
        C := CONCAT (CH,C);
```

Line 139 Col 1 Insert Indent A:NAME.PAS

```
VAL (C,RECNO,ERROR);
```

END;

IF FILESIZE(NAMEIN) = 0

THEN BEGIN

```
SCREEN_MAIN;
GOTOXY (15,12); WRITELN ('File is Empty !');
GOTO 31;
```

END

ELSE BEGIN

IF RECNO > FILESIZE (NAMEIN)

THEN BEGIN

```
GOTOXY(15,12);
WRITELN ('> End of File Encountered !');
GOTO 11;
```

END

ELSE BEGIN

WHILE NOT EOF(NAMEIN) DO

BEGIN

```
IF RECNO (>) 1
    THEN SEEK (NAMEIN,RECNO-1)
    ELSE RESET (NAMEIN);
READ (NAMEIN,INREC);
SCREEN_APPEND;
GOTOXY (15,10); WRITE(RECNO);
```

```

Line 162 Col 1 Insert Indent A:NAME.PAS
      GOTOXY (10,12); WRITE(INREC.CODE);
      GOTOXY (10,13); WRITE(INREC.NAME);
      GOTOXY (10,14); WRITE(INREC.UNIT);
      GOTOXY (10,15); WRITE(INREC.CLASS);
      SEEK (NAMEIN,RECNO-1);
      GOTOXY (10,12);
      WHILE NOT KEYSTAT(CH) DO BEGIN END;
      IF NOT (ORD(CH)=17)
        THEN BEGIN
          WRITE (CH); GOTOXY (11,12);
          READ(CV[RECNO]); CLREOL;
          CV[RECNO] := CONCAT(CH,CV[RECNO]);
          GOTOXY (10,13);
          READ (NV[RECNO]); CLREOL;
          GOTOXY (10,14);
          READ (BV[RECNO]); CLREOL;
          GOTOXY (10,15);
          READ (KV[RECNO]); CLREOL;
          1 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
          IF NOT (CH IN COMMANDSET)
            THEN BEGIN
              BEEP; GOTO 1;
            END
          ELSE CASE ORD(CH) OF

```

```

Line 185 Col 1 Insert Indent A:NAME.PAS
      59 : BEGIN
          WITH INREC DO
            BEGIN
              REC_NO := RECNO;
              CODE   := CV[RECNO];
              NAME   := NV[RECNO];
              UNIT   := BV[RECNO];
              CLASS  := KV[RECNO];
              NAMELENGTH := LENGTH(NV[RE
            END;
          WRITE(NAMEIN,INREC);
          SCREEN_APPEND;
        END;
      27 : BEGIN
          SCREEN_APPEND;
          RECNO := RECNO-1;
        END;
      17 : SEEK (NAMEIN,FILESIZE(NAMEIN)
        END;
      END
      ELSE SEEK (NAMEIN,FILESIZE(NAMEIN));
      RECNO := RECNO + 1;
    END;
  END;

```

```

Line 208 Col 1 Insert Indent A:NAME.PAS
      END;
21 : RECNO := 1;
      SCREEN_MAIN;
31 : END;

```

```

'A' :BEGIN
      RECNO := FILESIZE(NAMEIN);
      SEEK (NAMEIN,RECNO);
      SCREEN_APPEND;
      EXIT := FALSE;
      REPEAT
27 :GOTOXY(15,10);
      WRITE(RECNO +1);
      GOTOXY(10,12);
      WHILE NOT KEYSTAT(CH) DO BEGIN END;
      IF NOT (ORD(CH) = 17)
      THEN BEGIN
          RECNO := RECNO + 1;
          GOTOXY(10,12);
          WRITE(CH);
          READ(C);
          CV[RECNO] := CONCAT (CH,C);
          GOTOXY(10,13);

```

```

Line 231 Col 1 Insert Indent A:NAME.PAS
      READ(NV[RECNO]);
      GOTOXY(10,14);
      READ(BV[RECNO]);
      GOTOXY(10,15);
      READ(KV[RECNO]);
41 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
      IF NOT (CH IN COMMANDSET)
      THEN BEGIN
          BEEP;
          GOTO 1;
      END
      ELSE CASE ORD(CH) OF
59 : BEGIN
          WITH INREC DO
          BEGIN
              REC_NO := RECNO;
              CODE := CV[RECNO];
              NAME := NV[RECNO];
              UNIT := BV[RECNO];
              CLASS := KV[RECNO];
              NAMELENGTH := LENGTH(NV[RECNO]);
          END;
          WRITE(NAMEIN,INREC);
          SCREEN_APPEND;

```

Line 254 Col 1 Insert Indent A:NAME.PAS

```

                END;
                27 : BEGIN
                    SCREEN_APPEND;
                    RECNO := RECNO-1;
                    GOTO 27;
                END;
                17 : EXIT := TRUE;
                END;
            END
        ELSE EXIT := TRUE;
    UNTIL EXIT ;
    SCREEN_MAIN;
    RECNO := 1;
END;
END;
END;

```

BEGIN

```

SCREEN_MAIN;
ASSIGN (NAMEIN, 'A:NAME.DAT');
QUIT := FALSE;
REPEAT
    WHILE NOT KEYSTAT(CH) DO BEGIN END;
    IF CH IN COMMANDSET

```

Line 277 Col 1 Insert Indent A:NAME.PAS

```

    THEN CASE ORD(CH) OF
        1 : BEGIN
            BRANCH := 'A';
            RESET(NAMEIN);
            PROC_EDIT(NAMEIN, INREC, BRANCH);
        END;
        3 : BEGIN
            BRANCH := 'A';
            REWRITE (NAMEIN);
            PROC_EDIT(NAMEIN, INREC, BRANCH);
        END;
        5 : BEGIN
            BRANCH := 'E';
            RESET (NAMEIN);
            PROC_EDIT(NAMEIN, INREC, BRANCH);
        END;
        12 : BEGIN
            RESET(NAMEIN);
            RECNO := 1;
            GOTOXY(0,10);WRITELN;
            WRITELN (' ' :5, 'NO', ' ' :5, 'CODE', ' ' :10, 'NAME',
                ' ' :12, 'UNIT', ' ' :7, 'CLASS');WRITELN;
            WHILE NOT EOF(NAMEIN) DO
                BEGIN

```

```

Line 300 Col 1 Insert Indent A:NAME.PAS
      READ(NAMEIN,INREC);
      WITH INREC DO BEGIN
      NL := NAMELENGTH+5;
      WRITELN(REC_NO:7,CODE:9,NAME:NL,
              ' ':26-NL,UNIT:4,CLASS:10);END;
      RECNO := RECNO+1;
      END;
      WRITELN;WRITE('NAME File':20,
                  'Number of Records : ':30,FILESIZE(NAMEIN));
      END;
16 : BEGIN
      GOTOXY (10,10);WRITELN(') Printing ...');CLREOL;
      RESET(NAMEIN);
      RECNO := 1;
      WRITELN(LST,'NO':15,'CODE':12,' ':10,'NAME',
              'UNIT':30,'CLASS':12);
      WRITELN(LST,'--':15,'----':12,' ':10,'----',
              '-----':30,'-----':12);WRITELN(LST);
      WHILE NOT EOF(NAMEIN) DO
      BEGIN
      READ(NAMEIN,INREC);
      WITH INREC DO BEGIN
      NL := NAMELENGTH+10;
      WRITELN(LST,REC_NO:15,CODE:12,NAME:NL,

```

```

Line 335 Col 1 Insert Indent A:NAME.PAS
              ' ':34-NL,UNIT:10,CLASS:10);END;
      WRITELN(LST);
      RECNO := RECNO+1;
      END;
      WRITELN(LST);WRITELN(LST,' ':15,'NAME File',
                  'Number of Records : ':30,FILESIZE(NAMEIN));
      GOTOXY (10,10);WRITELN(') Press ALT-PrtSc');CLREOL;
      END;
17 : QUIT := TRUE;
      END
      ELSE BEGIN;
      BEEP;
      END;
      UNTIL QUIT;
      CLOSE (NAMEIN);
      END.

```

Logged drive: A
Active directory: \

Work file: A:NAME.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 12751 bytes
Free: 49884 bytes

)

Compiling
 328 lines

Code: 0108 paragraphs (4224 bytes), 0075 paragraphs free
Data: 0006 paragraphs (96 bytes), 00E1 paragraphs free
Stack/Heap: 2FF5 paragraphs (196432 bytes)

)

Line 1 Col 1 Insert Indent A:DEPTNAME.PAS

```
PROGRAM DEPARTMENTNAMEDATA;
```

```
{$V-}
```

```
TYPE
```

```
  STRING4 = STRING[4];
```

```
  STRING20 = STRING[20];
```

```
  INPUTRECORD = RECORD
```

```
    REC_NO : INTEGER;
```

```
    DEPTNAME : STRING20;
```

```
    NAMELENGTH: INTEGER;
```

```
  END;
```

```
  RECFILE = FILE OF INPUTRECORD;
```

```
VAR
```

```
  RECNO : INTEGER;
```

```
  CH : CHAR;
```

```
  BRANCH : CHAR;
```

```
  NL : INTEGER;
```

```
  QUIT : BOOLEAN;
```

```
  INREC : INPUTRECORD;
```

```
  DEPTNAMEFILE : FILE OF INPUTRECORD;
```

```
CONST
```

```
  COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];
```

```
FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;
```

Line 47 Col 1 Insert Indent A:DEPTNAME.PAS

```
TYPE REG_PACK = RECORD
```

```
  AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
```

```
  END;
```

```
VAR REGISTERS : REG_PACK;
```

```
BEGIN
```

```
  REGISTERS.AX := $0600;
```

```
  REGISTERS.DX := 255;
```

```
  MSDOS(REGISTERS);
```

```
  CH :=CHR(REGISTERS.AX);
```

```
  KEYSTAT := CH (<) CHR(0)
```

```
END;
```

```
PROCEDURE BEEP;
```

```
BEGIN
```

```
  WRITE(CHR(7));
```

```
END;
```

Line 70 Col 1 Insert Indent A:DEPTNAME.PAS

```

PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** ');
  GOTOXY (12,2); WRITELN ('! CTRL-C : CLEAR ');
  GOTOXY (12,3); WRITELN ('! CTRL-A : APPEND ');
  GOTOXY (12,4); WRITELN ('! CTRL-E : EDIT ');
  GOTOXY (12,5); WRITELN ('! CTRL-Q : QUIT ');
  GOTOXY (12,6); WRITELN ('! CTRL-L : LIST ');
  GOTOXY (12,7); WRITELN ('! CTRL-P : PRINT ');
END;
```

PROCEDURE SCREEN_APPEND;

```

BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** ');
  GOTOXY (12,2); WRITELN ('! F1 : SAVE ');
  GOTOXY (12,3); WRITELN ('! ESC : ESCAPE ');
  GOTOXY (12,4); WRITELN ('! CTRL-Q : QUIT ');
  GOTOXY (5,10);
  WRITELN('REC_NO :');WRITELN;
  WRITELN('NAME :');
END;
```

Line 93 Col 1 Insert Indent A:DEPTNAME.PAS

```

PROCEDURE PROC_EDIT (VAR DEPTNAMEFILE : RECFILE;
  VAR INREC : INPUTRECORD;
  VAR BRANCH : CHAR);
```

LABEL 1,11,21,31,41,27;

TYPE

```

  INDEX = 1..5;
  STRING4 = STRING[4];
  STRING20 = STRING[20];
  RECNOVECTOR = ARRAY [INDEX] OF INTEGER;
  DEPTNAMEVECTOR = ARRAY [INDEX] OF STRING20;
  LENGTHVECTOR = ARRAY [INDEX] OF INTEGER;
```

VAR

```

  RV : RECNOVECTOR;
  DNV : DEPTNAMEVECTOR;
  LV : LENGTHVECTOR;
  CH : CHAR;
  C : STRING20;
  ERROR : INTEGER;
  EXIT : BOOLEAN;
```

CONST

```

  COMMANDSET : SET OF CHAR = [#17,#27,#59];
```

```

Line 116 Col 1 Insert Indent A:DEPTNAME.PAS
BEGIN
CASE BRANCH OF
'E' :BEGIN
EXIT := FALSE;
CLRSCR;
GOTOXY(10,10);
WRITELN('FIND RECNO = ');
11 : GOTOXY (23,10);CLREOL;
WHILE NOT KEYSTAT (CH) DO BEGIN END;
IF ((ORD(CH)=27) OR (ORD(CH)=17))
THEN GOTO 21
ELSE BEGIN
WRITE (CH);
READ(C);
C := CONCAT (CH,C);
VAL (C,RECNO,ERROR);
END;
IF FILESIZE(DEPTNAMEFILE) = 0
THEN BEGIN
SCREEN_MAIN;
GOTOXY (15,12); WRITELN ('File is Empty !');
GOTO 31;
END
ELSE BEGIN

Line 139 Col 1 Insert Indent A:DEPTNAME.PAS
IF RECNO > FILESIZE (DEPTNAMEFILE)
THEN BEGIN
GOTOXY(15,12);
WRITELN (' End of File Encountered !');
GOTO 11;
END
ELSE BEGIN
WHILE NOT EOF(DEPTNAMEFILE) DO
BEGIN
IF RECNO <> 1
THEN SEEK (DEPTNAMEFILE,RECNO-1)
ELSE RESET (DEPTNAMEFILE);
READ (DEPTNAMEFILE,INREC);
SCREEN_APPEND;
GOTOXY (15,10); WRITE(RECNO);
GOTOXY (10,12); WRITE(INREC.DEPTNAME);
SEEK (DEPTNAMEFILE,RECNO-1);
GOTOXY (10,12);
WHILE NOT KEYSTAT(CH) DO BEGIN END;
IF NOT (ORD(CH)=17)
THEN BEGIN
WRITE (CH); GOTOXY (11,12);
READ(DNV[RECNO]);CLREOL;
DNV[RECNO] := CONCAT(CH,DNV[RECNO]);

```

```

Line 162 Col 1 Insert Indent A:DEPTNAME.PAS
      1 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
      IF NOT (CH IN COMMANDSET)
      THEN BEGIN
          BEEP; GOTO 1;
      END
      ELSE CASE ORD(CH) OF
          59 : BEGIN
              WITH INREC DO
              BEGIN
                  REC_NO := RECNO;
                  DEPTNAME := DNV[RECNO];
                  NAMELENGTH := LENGTH(DNV[R
              END;
              WRITE(DEPTNAMEFILE,INREC);
              SCREEN_APPEND;
          END;
          27 : BEGIN
              SCREEN_APPEND;
              RECNO := RECNO-1;
          END;
      WRITE (CH); GOTOXY (11,12);
      END;
      END
      ELSE SEEK (DEPTNAMEFILE,FILESIZE(DEPTNAMEFILE))

```

```

Line 177 Col 1 Insert Indent A:DEPTNAME.PAS
      RECNO := RECNO + 1;
      END;
      END;
      END;
21 : RECNO := 1;
      SCREEN_MAIN;
31 : END;

```

```

'A' :BEGIN
      RECNO := FILESIZE(DEPTNAMEFILE);
      SEEK (DEPTNAMEFILE,RECNO);
      SCREEN_APPEND;
      EXIT := FALSE;
      REPEAT
          27 :GOTOXY(15,10);
          WRITE(RECNO +1);
          GOTOXY(10,12);
          WHILE NOT KEYSTAT(CH) DO BEGIN END;
          IF NOT (ORD(CH) = 17)
          THEN BEGIN
              RECNO := RECNO + 1;
              GOTOXY(10,12);
              WRITE(CH);
          END;
      UNTIL EXIT;
      END;

```

```

Line 200 Col 1 Insert Indent A:DEPTNAME.PAS
      READ(C);
      DNV[RECNO] := CONCAT (CH,C);
      41 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
          IF NOT (CH IN COMMANDSET)
              THEN BEGIN
                  BEEP;
                  GOTO 1;
              END
          ELSE CASE ORD(CH) OF
              59 : BEGIN
                  WITH INREC DO
                  BEGIN
                      REC_NO := RECNO;
                      DEPTNAME := DNV[RECNO];
                      NAMELENGTH := LENGTH(DNV[RECNO]);
                  END;
                  WRITE(DEPTNAMEFILE,INREC);
                  SCREEN_APPEND;
              END;
              27 : BEGIN
                  SCREEN_APPEND;
                  RECNO := RECNO-1;
                  GOTO 27;
              END;
          END;

```

```

Line 226 Col 1 Insert Indent A:DEPTNAME.PAS
      17 : EXIT := TRUE;
      END;

```

```

      END
      ELSE EXIT := TRUE;
      UNTIL EXIT ;
      SCREEN_MAIN;
      RECNO := 1;
END;
END;
END;

BEGIN
  SCREEN_MAIN;
  ASSIGN (DEPTNAMEFILE,'A:DEPTNAME.DAT');
  QUIT := FALSE;
  REPEAT
    WHILE NOT KEYSTAT(CH) DO BEGIN END;
    IF CH IN COMMANDSET
      THEN CASE ORD(CH) OF
          1 : BEGIN
              BRANCH := 'A';
              RESET(DEPTNAMEFILE);
              PROC_EDIT(DEPTNAMEFILE,INREC,BRANCH);
            END;

```

```

Line 242 Col 1 Insert Indent A:DEPTNAME.PAS
  3 : BEGIN
      BRANCH := 'A';
      REWRITE (DEPTNAMEFILE);
      PROC_EDIT(DEPTNAMEFILE,INREC,BRANCH);
      END;
  5 : BEGIN
      BRANCH := 'E';
      RESET (DEPTNAMEFILE);
      PROC_EDIT(DEPTNAMEFILE,INREC,BRANCH);
      END;
 12 : BEGIN
      RESET(DEPTNAMEFILE);
      RECNO := 1;
      GOTOXY(5,10);WRITELN;
      WRITELN (' ' :5,'NO', ' ' :10,'PRODUCTION UNIT');WRITELN;
      WHILE NOT EOF(DEPTNAMEFILE) DO
      BEGIN
        READ(DEPTNAMEFILE,INREC);
        WITH INREC DO BEGIN
          NL := NAMELENGTH+5;
          WRITELN(REC_NO:7,DEPTNAME:NL); END;
          RECNO := RECNO+1;
        END;
      WRITELN;WRITELN;WRITE('DEPTNAME File ' :20,
Line 266 Col 1 Insert Indent A:DEPTNAME.PAS
      'Number of Records : ' :30,FILESIZE(DEPTNAMEFILE));
      END;
 16 : BEGIN
      GOTOXY(10,10);WRITELN('>> Printing ...');CLREOL;
      RESET(DEPTNAMEFILE);
      RECNO := 1;
      WRITELN(LST,'NO':15,'PRODUCTION UNIT':25);
      WRITELN(LST,'--':15,'-----':25);WRITELN(LST)
      END;
      BEGIN
        READ(DEPTNAMEFILE,INREC);
        WITH INREC DO BEGIN
          NL := NAMELENGTH+10;
          WRITELN(LST,REC_NO:15,DEPTNAME:NL);END;
          WRITELN(LST);
          RECNO := RECNO+1;
        END;
      WRITELN(LST);WRITELN(LST,' ' :15,'DEPTNAME File',
        'Number of Records : ' :30,FILESIZE(DEPTNAMEFILE));
      GOTOXY(10,10);WRITELN('>> Press ALT-PrtSc');
      END;
 17 : QUIT := TRUE;
      END

```

```

Line 301 Col 1 Insert Indent A:DEPTNAME.PAS
      ELSE BEGIN;
            BEEP;
            END;
UNTIL QUIT;
CLOSE (DEPTNAMEFILE);
END.

```

Logged drive: A
Active directory: \

Work file: A:DEPTNAME.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 11127 bytes
Free: 51508 bytes

)

Compiling
295 lines

Code: 0005 paragraphs (3408 bytes), 0CA8 paragraphs free
Data: 0005 paragraphs (80 bytes), 0FE2 paragraphs free
Stack/Heap: 308F paragraphs (198896 bytes)

)

```

Line 1 Col 1 Insert Indent A:PERNAME.PAS
PROGRAM PERIODNAMEDATA;
{$V-}
TYPE
  STRING4 = STRING[4];
  STRING20 = STRING[20];
  INPUTRECORD = RECORD
    REC_NO : INTEGER;
    PERNAME : STRING20;
    NAMELENGTH: INTEGER;
  END;
  RECFILE = FILE OF INPUTRECORD;
VAR
  RECNO : INTEGER;
  CH : CHAR;
  BRANCH : CHAR;
  NL : INTEGER;
  QUIT : BOOLEAN;
  INREC : INPUTRECORD;
  PERNAMEFILE : FILE OF INPUTRECORD;
CONST
  COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];

```

```
FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;
```

```

Line 47 Col 1 Insert Indent A:PERNAME.PAS
TYPE REG_PACK = RECORD
  AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
END;
VAR REGISTERS : REG_PACK;
BEGIN
  REGISTERS.AX := $0600;
  REGISTERS.DX := 255;
  MSDOS(REGISTERS);
  CH :=CHR(REGISTERS.AX);
  KEYSTAT := CH (<) CHR(0)
END;
```

```

PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;
```


Line 70 Col 1 Insert Indent A:PERNAME.PAS
 PROCEDURE SCREEN_MAIN;

```
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** ');
  GOTOXY (12,2); WRITELN ('! CTRL-C : CLEAR ');
  GOTOXY (12,3); WRITELN ('! CTRL-A : APPEND ');
  GOTOXY (12,4); WRITELN ('! CTRL-E : EDIT ');
  GOTOXY (12,5); WRITELN ('! CTRL-Q : QUIT ');
  GOTOXY (12,6); WRITELN ('! CTRL-L : LIST ');
  GOTOXY (12,7); WRITELN ('! CTRL-P : PRINT ');
END;
```

PROCEDURE SCREEN_APPEND;

```
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** ');
  GOTOXY (12,2); WRITELN ('! F1 : SAVE ');
  GOTOXY (12,3); WRITELN ('! ESC : ESCAPE ');
  GOTOXY (12,4); WRITELN ('! CTRL-Q : QUIT ');
  GOTOXY (5,10);
  WRITELN('REC_NO :');WRITELN;
  WRITELN('NAME :');
END;
```

Line 74 Col 1 Insert Indent A:PERNAME.PAS

```
PROCEDURE PROC_EDIT (VAR PERNAMEFILE : RECFILE;
  VAR INREC : INPUTRECORD;
  VAR BRANCH : CHAR);
```

LABEL 1,11,21,31,41,27;

TYPE

```
INDEX = 1..20;
STRING4 = STRING[4];
STRING20 = STRING[20];
RECNOVECTOR = ARRAY [INDEX] OF INTEGER;
PERNAMEVECTOR = ARRAY [INDEX] OF STRING20;
LENGTHVECTOR = ARRAY [INDEX] OF INTEGER;
```

VAR

```
RV : RECNOVECTOR;
PNV : PERNAMEVECTOR;
LV : LENGTHVECTOR;
CH : CHAR;
C : STRING20;
ERROR : INTEGER;
EXIT : BOOLEAN;
```

CONST

```
COMMANDSET : SET OF CHAR = [#17,#27,#59];
```

```

Line 120 Col 22 Insert Indent A:PERNAME.PAS
BEGIN
CASE BRANCH OF
'E' :BEGIN
EXIT := FALSE;
CLRSCR;
GOTOXY(10,10);
WRITELN('FIND RECNO = ');
11 : GOTOXY (23,10);CLREOL;
WHILE NOT KEYSTAT (CH) DO BEGIN END;
IF ((ORD(CH)=27) OR (ORD(CH)=17))
THEN GOTO 21
ELSE BEGIN
WRITE (CH);
READ(C);
C := CONCAT (CH,C);
VAL (C,RECNO,ERROR);
END;
IF FILESIZE(PERNAMEFILE) = 0
THEN BEGIN
SCREEN_MAIN;
GOTOXY (15,12); WRITELN ('File is Empty !');
GOTO 31;
END
ELSE BEGIN

```

```

Line 143 Col 22 Insert Indent A:PERNAME.PAS
IF RECNO > FILESIZE (PERNAMEFILE)
THEN BEGIN
GOTOXY(15,12);
WRITELN ('> End of File Encountered !');
GOTO 11;
END
ELSE BEGIN
WHILE NOT EOF(PERNAMEFILE) DO
BEGIN
IF RECNO (<) 1
THEN SEEK (PERNAMEFILE,RECNO-1)
ELSE RESET (PERNAMEFILE);
READ (PERNAMEFILE,INREC);
SCREEN_APPEND;
GOTOXY (15,10); WRITE(RECNO);
GOTOXY (10,12); WRITE(INREC.PERNAME);
SEEK (PERNAMEFILE,RECNO-1);
GOTOXY (10,12);
WHILE NOT KEYSTAT(CH) DO BEGIN END;
IF NOT (ORD(CH)=17)
THEN BEGIN
WRITE (CH); GOTOXY (11,12);
READ(PNV[RECNO]);CLREOL;
PNV[RECNO] := CONCAT(CH,PNV[RECNO]);

```

```

Line 166 Col 22 Insert Indent A:PERNAME.PAS
      1 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
      IF NOT (CH IN COMMANDSET)
      THEN BEGIN
          BEEP; GOTO 1;
          END
      ELSE CASE ORD(CH) OF
          59 : BEGIN
              WITH INREC DO
              BEGIN
                  REC_NO := RECNO;
                  PERNAME := PNV[RECNO];
                  NAMELENGTH := LENGTH(PNV[R
                  END;
                  WRITE(PERNAMEFILE,INREC);
                  SCREEN_APPEND;
              END;
          27 : BEGIN
              SCREEN_APPEND;
              RECNO := RECNO-1;
              END;
          17 : SEEK (PERNAMEFILE,FILESIZE(P
              END;
      END
      ELSE SEEK (PERNAMEFILE,FILESIZE(PERNAMEFILE));

```

```

Line 189 Col 22 Insert Indent A:PERNAME.PAS
      RECNO := RECNO + 1;
      END;
      END;
      21 : RECNO := 1;
      SCREEN_MAIN;
      31 : END;

```

```

'A' :BEGIN
      RECNO := FILESIZE(PERNAMEFILE);
      SEEK (PERNAMEFILE,RECNO);
      SCREEN_APPEND;
      EXIT := FALSE;
      REPEAT
          27 :GOTOXY(15,10);
          WRITE(RECNO +1);
          GOTOXY(10,12);
          WHILE NOT KEYSTAT(CH) DO BEGIN END;
          IF NOT (ORD(CH) = 17)
          THEN BEGIN
              RECNO := RECNO + 1;
              GOTOXY(10,12);
              WRITE(CH);
          END;
      UNTIL FALSE;

```

```

Line 212 Col 22 Insert Indent A:PERNAME.PAS
    READ(C);
    PNV[RECNO] := CONCAT (CH,C);
    41 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
        IF NOT (CH IN COMMANDSET)
            THEN BEGIN
                BEEP;
                GOTO 1;
            END
        ELSE CASE ORD(CH) OF
            59 : BEGIN
                WITH INREC DO
                BEGIN
                    REC_NO := RECNO;
                    PERNAME := PNV[RECNO];
                    NAMELENGTH := LENGTH(PNV[RECNO]);
                END;
                WRITE(PERNAMEFILE,INREC);
                SCREEN_APPEND;
            END;
            27 : BEGIN
                SCREEN_APPEND;
                RECNO := RECNO-1;
                GOTO 27;
            END;
        END;

```

```

Line 230 Col 1 Insert Indent A:PERNAME.PAS
    17 : EXIT := TRUE;
    END;

```

```

    END
    ELSE EXIT := TRUE;
    UNTIL EXIT ;
    SCREEN_MAIN;
    RECNO := 1;
END;
END;
END;

```

```

BEGIN
    SCREEN_MAIN;
    ASSIGN (PERNAMEFILE,'A:PERNAME.DAT');
    QUIT := FALSE;
    REPEAT
        WHILE NOT KEYSTAT(CH) DO BEGIN END;
        IF CH IN COMMANDSET
            THEN CASE ORD(CH) OF
                1 : BEGIN
                    BRANCH := 'A';
                    RESET(PERNAMEFILE);
                END;
            END;
    UNTIL QUIT;
END;

```

```

Line 236 Col 1 Insert Indent A:PERNAME.PAS
PROC_EDIT(PERNAMEFILE,INREC,BRANCH);
END;
3 : BEGIN
  BRANCH := 'A';
  REWRITE (PERNAMEFILE);
  PROC_EDIT(PERNAMEFILE,INREC,BRANCH);
END;
5 : BEGIN
  BRANCH := 'E';
  RESET (PERNAMEFILE);
  PROC_EDIT(PERNAMEFILE,INREC,BRANCH);
END;
12 : BEGIN
  RESET(PERNAMEFILE);
  RECNO := 1;
  GOTOXY(0,11);WRITELN;
  WRITELN (' :5,'NO', ' :10,'PERIOD NAME');WRITELN;
  WHILE NOT EOF(PERNAMEFILE) DO
  BEGIN
    READ(PERNAMEFILE,INREC);
    WITH INREC DO BEGIN
      NL := NAMELENGTH+5;
      WRITELN(REC_NO:7,' :8,PERNAME:NL); END;
      RECNO := RECNO+1;
    END;
  END;

Line 259 Col 1 Insert Indent A:PERNAME.PAS
END;
WRITELN;WRITE('PERNAME File ':20,
  'Number of Records : ':30,FILESIZE(PERNAMEFILE));
END;
16 : BEGIN
  GOTOXY(10,10);WRITELN(') Printing ...');CLREOL;
  RESET(PERNAMEFILE);
  RECNO := 1;
  WRITELN(LST,'NO':15,' :8,'PERIOD NAME');
  WRITELN(LST,'--':15,' :8,'-----');WRITELN(LST);
  WHILE NOT EOF(PERNAMEFILE) DO
  BEGIN
    READ(PERNAMEFILE,INREC);
    WITH INREC DO BEGIN
      NL := NAMELENGTH+10;
      WRITELN(LST,REC_NO:15,PERNAME:NL);END;
      WRITELN(LST);
      RECNO := RECNO+1;
    END;
  END;
  WRITELN(LST);WRITELN(LST,' :15,'PERNAME File',
    'Number of Records : ':30,FILESIZE(PERNAMEFILE));
  GOTOXY(10,10);WRITELN(') Press ALT-PrtSC');
END;
17 : QUIT := TRUE;

```

)
Line 283 Col 1 Insert Indent A:PERNAME.PAS

END

ELSE BEGIN;

BEEP;

END;

UNTIL QUIT;

CLOSE (PERNAMEFILE);

END.

Logged drive: A

Active directory: \

Work file: A:PERNAME.PAS

Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 11062 bytes

Free: 51573 bytes

)
Compiling

289 lines

Code: 00D6 paragraphs (3424 bytes), 0CA7 paragraphs free

Data: 0005 paragraphs (80 bytes), 0FE2 paragraphs free

Stack/Heap: 3092 paragraphs (198944 bytes)

)

Line 50 Col 1 Insert Indent A:ABCIN.PAS

```
PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;
```

```
PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('| ***** MENU **** |');
  GOTOXY (12,2); WRITELN ('| CTRL-C : CLEAR |');
  GOTOXY (12,3); WRITELN ('| CTRL-A : APPEND |');
  GOTOXY (12,4); WRITELN ('| CTRL-E : EDIT |');
  GOTOXY (12,5); WRITELN ('| CTRL-Q : QUIT |');
  GOTOXY (12,6); WRITELN ('| CTRL-L : LIST |');
  GOTOXY (12,7); WRITELN ('| CTRL-P : PRINT |');
  GOTOXY (12,8); WRITELN ('| CTRL-S : SORT |');
END;
```

Line 88 Col 1 Insert Indent A:ABCIN.PAS

```
PROCEDURE SCREEN_APPEND;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('| ***** MENU **** |');
  GOTOXY (12,2); WRITELN ('| F1 : SAVE |');
  GOTOXY (12,3); WRITELN ('| ESC : ESCAPE |');
  GOTOXY (12,4); WRITELN ('| CTRL-Q : QUIT |');
  GOTOXY (5,10);
  WRITELN('REC_NO :');WRITELN;
  WRITELN('CODE :');
  WRITELN('PRICE :');
  WRITELN('USAGE :');
END;
```

```
PROCEDURE PROC_EDIT (VAR NAMEREC: NAMERECORD;
  VAR NAMEIN : NAMEFILE;
  VAR INREC : INPUTRECORD;
  VAR ABCIN : RECFILE;
  VAR BRANCH : CHAR);
```

LABEL 1,11,21,31,41,27;

Line 111 Col 1 Insert Indent A:ABCIN.PAS

TYPE

```
INDEX = 1..20;
STRING4 = STRING[4];
RECNOVECTOR = ARRAY [INDEX] OF INTEGER;
CODEVECTOR = ARRAY [INDEX] OF STRING4;
PRICEVECTOR = ARRAY [INDEX] OF REAL;
USAGEVECTOR = ARRAY [INDEX] OF REAL;
```

VAR

```
RV : RECNOVECTOR;
CV : CODEVECTOR;
PV : PRICEVECTOR;
UV : USAGEVECTOR;
CH : CHAR;
C,COD : STRING4;
I : INTEGER;
RECNO : INTEGER;
EXIT : BOOLEAN;
ERROR : INTEGER;
```

CONST

```
COMMANDSET : SET OF CHAR = [#17,#27,#59];
```

BEGIN

```
CASE BRANCH OF
'E' :BEGIN
    EXIT := FALSE;
```

Line 134 Col 1 Insert Indent A:ABCIN.PAS

```
CLRSCR;
GOTOXY(10,10);
WRITELN('FIND RECNO = ');
11 : GOTOXY(23,10);
WHILE NOT KEYSTAT(CH) DO BEGIN END;
IF ((ORD(CH)=27) OR (ORD(CH)=17))
    THEN GOTO 21
    ELSE BEGIN
        WRITE (CH);
        READ(C);
        C := CONCAT (CH,C);
        VAL (C,RECNO,ERROR);
    END;
IF FILESIZE(ABCIN) = 0
    THEN BEGIN
        SCREEN_MAIN;
        GOTOXY (15,12); WRITELN ('File is Empty !');
        GOTO 31;
    END
    ELSE BEGIN
        IF RECNO > FILESIZE (ABCIN)
            THEN BEGIN
                GOTOXY(15,12);
                WRITELN (' End of File Encountered !');
```



```

Line 157 Col 1 Insert Indent A:ABCIN.PAS
      GOTO 11;
      END
ELSE BEGIN
      WHILE NOT EOF (ABCIN) DO
      BEGIN
        IF RECNO (>) 1 THEN SEEK (ABCIN,RECNO-1)
          ELSE RESET (ABCIN);
        READ (ABCIN,INREC);
        SCREEN_APPEND;
        GOTOXY (15,10); WRITE(RECNO);
        GOTOXY (10,12); WRITE(INREC.CODE);
        GOTOXY (10,13); WRITE(INREC.PRICE);
        GOTOXY (10,14); WRITE(INREC.USAGE);
        SEEK (ABCIN,RECNO-1);
        GOTOXY (10,12);
        WHILE NOT KEYSTAT(CH) DO BEGIN END;
        IF NOT (ORD(CH)=17)
          THEN BEGIN
            WRITE (CH);
            GOTOXY (11,12);
            READ (CV[RECNO]); CLREOL;
            CV[RECNO] := CONCAT(CH,CV[RECNO]);
            GOTOXY (10,13);
            READ (PV[RECNO]); CLREOL;
          END;
      END;
      GOTOXY (10,14);
      READ (UV[RECNO]); CLREOL;
      1 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
      IF NOT (CH IN COMMANDSET)
        THEN BEGIN
          BEEP; GOTO 1;
        END
      ELSE CASE ORD(CH) OF
        59 : BEGIN
          WITH INREC DO
          BEGIN
            REC_NO := RECNO;
            CODE := CV[RECNO];
            PRICE := PV[RECNO];
            USAGE := UV[RECNO];
          END;
          WRITE(ABCIN,INREC);
          SCREEN_APPEND;
        END;
        27 : BEGIN
          SCREEN_APPEND;
          RECNO := RECNO-1;
        END;
        17 : SEEK (ABCIN,FILESIZE(ABCIN))
      END;

```

```

Line 213 Col 1 Insert Indent A:ABCIN.PAS
                        END;
                        END
ELSE SEEK (ABCIN,FILESIZE(ABCIN));
RECNO := RECNO + 1;
END;
END;
END;
21 : RECNO := 1;
SCREEN_MAIN;
RECNO := 1;
31 : END;

```

```

'A' : BEGIN
      MAXREC := FILESIZE(NAMEIN)-FILESIZE(ABCIN);
      RECNO := FILESIZE(ABCIN);
      SEEK (ABCIN,RECNO);
      FOR I := 1 TO MAXREC DO
      BEGIN
        WITH INREC DO
        BEGIN
          CODE := '....';
          PRICE := 0;
          USAGE := 0;
        END;
      END;

```

```

Line 236 Col 1 Insert Indent A:ABCIN.PAS
      WRITE (ABCIN,INREC);
      END;
      SCREEN_MAIN;
      GOTOXY (15,12);
      WRITELN ('> ',(MAXREC),' Records Added');
      END;

```

```

'S' : BEGIN
      GOTOXY(15,12);WRITELN('>> Sorting ...');CLREOL;
      RESET (NAMEIN);
      RESET (ABCIN);
      I := 1;
      J := 1;
      WHILE NOT EOF (ABCIN) DO
      BEGIN
        READ (ABCIN,INREC);
        WITH INREC DO
        BEGIN
          RV[I] := REC_NO;
          CV[I] := CODE;
          PV[I] := PRICE;
          UV[I] := USAGE;
        END;
        I := I+1;
      END;

```

Line 259 Col 1 Insert Indent A:ABCIN.PAS

```

MAXREC := FILESIZE(ABCIN);
REWRITE(ABCIN);
WHILE NOT EOF (NAMEIN) DO
BEGIN
  READ (NAMEIN,NAMEREC);
  RECNO := NAMEREC.REC_NO;
  COD := NAMEREC.CODE;
  FOR I := 1 TO MAXREC DO
  BEGIN
    IF COD = CV[I]
      THEN BEGIN
        WITH INREC DO
        BEGIN
          REC_NO := RECNO;
          CODE := CV[I];
          PRICE := PV[I];
          USAGE := UV[I];
        END;
        WRITE (ABCIN,INREC);
      END;
    END;
    J := J+1;
  END;
  GOTOXY(15,12);WRITELN('>> Sort Completed.');
```

Line 273 Col 1 Insert Indent A:ABCIN.PAS

```

END;
END;
END;
```

BEGIN

```

SCREEN_MAIN;
ASSIGN (NAMEIN,'A:NAME.DAT');
ASSIGN (ABCIN,'A:ABCIN.DAT');
RESET(NAMEIN);
QUIT := FALSE;
REPEAT
  WHILE NOT KEYSTAT(CH) DO BEGIN END;
  IF CH IN COMMANDSET
    THEN CASE ORD(CH) OF
      1 : BEGIN
          RESET (ABCIN);
          BRANCH := 'A';
          PROC_EDIT(NAMEREC,NAMEIN,INREC,ABCIN,BRANCH);
        END;
```

```

Line 296 Col 1 Insert Indent A:ABCIN.PAS
  3 : BEGIN
      REWRITE (ABCIN);
      BRANCH := 'A';
      PROC_EDIT(NAMEREC,NAMEIN,INREC,ABCIN,BRANCH);
  END;
  5 : BEGIN
      BRANCH := 'E';
      RESET (ABCIN);
      PROC_EDIT(NAMEREC,NAMEIN,INREC,ABCIN,BRANCH);
  END;
 12 : BEGIN
      RESET(ABCIN);
      RECNO := 1;
      SCREEN_MAIN;
      GOTOXY(1,10);
      WRITELN (' :5,'NO', ' :5,'CODE', ' :5,
'PURCHASE PRICE', ' :5,'ANNUAL USAGE');
      WHILE NOT EOF(ABCIN) DO
      BEGIN
          READ(ABCIN,INREC);
          WITH INREC DO BEGIN
              WRITELN(REC_NO:7,CODE:9,PRICE:17:2,
                      USAGE:18:2);END;
          RECNO := RECNO+1;
      END;
  END;

```

```

Line 319 Col 1 Insert Indent A:ABCIN.PAS
      END;
      WRITELN;WRITE('ABCIN File':20,
'Number of Records : ':30,FILESIZE(ABCIN));
  END;
 16 : BEGIN
      GOTOXY(15,12);WRITELN(') Printing ...');CLREOL;
      RESET(ABCIN);
      RECNO := 1;
      WRITELN(LST);WRITELN(LST);
      WRITELN(LST,' :10,'ABCIN File');WRITELN(LST);
      WRITELN (LST,' :5,'NO', ' :5,'CODE', ' :5,
'PURCHASE PRICE', ' :5,'ANNUAL USAGE');
      WHILE NOT EOF(ABCIN) DO
      BEGIN
          READ(ABCIN,INREC);
          WITH INREC DO BEGIN
              WRITELN(LST,REC_NO:7,CODE:9,PRICE:14:2,
                      USAGE:18:2);END;
          RECNO := RECNO+1;
      END;
      WRITELN(LST);WRITE(LST,'ABCIN File':20,
'Number of Records : ':30,FILESIZE(ABCIN));
  END;
 17 : QUIT := TRUE;

```

```

Line 342 Col 1 Insert Indent A:ABCIN.PAS
      19 : BEGIN
            BRANCH := 'S';
            PROC_EDIT (NAMEREC,NAMEIN,INREC,ABCIN,BRANCH);
            END;
      END
    ELSE BEGIN;
      BEEP;
    END;
  UNTIL QUIT;
CLOSE (ABCIN);
CLOSE (NAMEIN);
END.

```

Logged drive: A
Active directory: \

Work file: A:ABCIN.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 12513 bytes
Free: 50122 bytes

)

Compiling
348 lines

Code: 010A paragraphs (4256 bytes), 0C73 paragraphs free
Data: 000B paragraphs (176 bytes), 0FDC paragraphs free
Stack/Heap: 2FFD paragraphs (196560 bytes)

)

Line 1 Col 1 Insert Indent A:INVIN.PAS

PROGRAM INVDATA;

{\$V-}

TYPE

STRING4 = STRING[4];

STRING20 = STRING[20];

NAMERECORD = RECORD

REC_NO : INTEGER;

CODE : STRING4;

NAME : STRING20;

UNIT : STRING4;

CLASS : STRING4;

NAMELENGTH : INTEGER;

END;

INPUTRECORD = RECORD

REC_NO : INTEGER;

CODE : STRING4;

REPLCOST : REAL;

LEADTIME : INTEGER;

LEADVARIANCE : REAL;

END;

NAMEFILE = FILE OF NAMERECORD;

RECFILE = FILE OF INPUTRECORD;

VAR

CH : CHAR;

Line 25 Col 1 Insert Indent A:INVIN.PAS

I,J : INTEGER;

RECNO : INTEGER;

BRANCH : CHAR;

QUIT : BOOLEAN;

MAXREC : INTEGER;

INREC : INPUTRECORD;

NAMEREC : NAMERECORD;

INVIN : FILE OF INPUTRECORD;

NAMEIN : FILE OF NAMERECORD;

CONST

COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17,#19];

FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;

TYPE REG_PACK = RECORD

AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;

END;

VAR REGISTERS : REG_PACK;

BEGIN

REGISTERS.AX := \$0600;

REGISTERS.DX := 255;

MSDOS(REGISTERS);

CH :=CHR(REGISTERS.AX);

KEYSTAT := CH (<) CHR(0)

END;

Line 49 Col 1 Insert Indent A:INVIN.PAS

```
PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;
```

```
PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('| ***** MENU ***** |');
  GOTOXY (12,2); WRITELN ('| CTRL-C : CLEAR      |');
  GOTOXY (12,3); WRITELN ('| CTRL-A : APPEND    |');
  GOTOXY (12,4); WRITELN ('| CTRL-E : EDIT      |');
  GOTOXY (12,5); WRITELN ('| CTRL-Q : QUIT      |');
  GOTOXY (12,6); WRITELN ('| CTRL-L : LIST      |');
  GOTOXY (12,7); WRITELN ('| CTRL-P : PRINT     |');
  GOTOXY (12,8); WRITELN ('| CTRL-S : SORT      |');
END;
```

Line 73 Col 1 Insert Indent A:INVIN.PAS

```
PROCEDURE SCREEN_APPEND;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('| ***** MENU ***** |');
  GOTOXY (12,2); WRITELN ('| F1      : SAVE      |');
  GOTOXY (12,3); WRITELN ('| ESC     : ESCAPE    |');
  GOTOXY (12,4); WRITELN ('| CTRL-Q  : QUIT     |');
  GOTOXY (5,10);
  WRITELN('REC_NO :');WRITELN;
  WRITELN('CODE      :');
  WRITELN('REPLCOST   :');
  WRITELN('LEADTIME   :');
  WRITELN('LEADVARIANCE :');
END;
```

```
PROCEDURE PROC_EDIT (VAR NAMEREC: NAMERECORD;
  VAR NAMEIN : NAMEFILE;
  VAR INREC  : INPUTRECORD;
  VAR INVIN  : RECFILE;
  VAR BRANCH : CHAR);
```

```
LABEL 1,11,21,31,41,27;
```

Line 97 Col 1 Insert Indent A:INVIN.PAS
TYPE

```
INDEX = 1..20;
STRING4 = STRING[4];
RECNOVECTOR      = ARRAY [INDEX] OF INTEGER;
CODEVECTOR       = ARRAY [INDEX] OF STRING4;
REPLCOSTVECTOR   = ARRAY [INDEX] OF REAL;
LEADTIMEVECTOR   = ARRAY [INDEX] OF INTEGER;
LEADVARIANCEVECTOR = ARRAY [INDEX] OF REAL;
```

VAR

```
RV : RECNOVECTOR;
WRITELN('LEADTIME      ');
RCV : REPLCOSTVECTOR;
LTV : LEADTIMEVECTOR;
LVV : LEADVARIANCEVECTOR;
CH : CHAR;
C,COD : STRING4;
I : INTEGER;
RECNO : INTEGER;
EXIT : BOOLEAN;
ERROR : INTEGER;
```

CONST

```
COMMANDSET : SET OF CHAR = [#17,#27,#59];
```

BEGIN

```
CASE BRANCH OF
```

Line 143 Col 20 Insert Indent A:INVIN.PAS
'E' :BEGIN

```
EXIT := FALSE;
CLRSCR;
GOTOXY(10,10);
WRITELN('FIND RECNO = ');
I1 : GOTOXY(23,10);
WHILE NOT KEYSTAT(CH) DO BEGIN END;
IF ((ORD(CH)=27) OR (ORD(CH)=17))
    THEN GOTO 21
    ELSE BEGIN
        WRITE (CH);
        READ(C);
        C := CONCAT (CH,C);
        VAL (C,RECNO,ERROR);
    END;
IF FILESIZE(INVIN) = 0
    THEN BEGIN
        SCREEN_MAIN;
        GOTOXY (15,12); WRITELN ('File is Empty !');
        GOTO 31;
    END
ELSE BEGIN
    IF RECNO > FILESIZE (INVIN)
```


Line 166 Col 20 Insert Indent A:INVIN.PAS

```

THEN BEGIN
  GOTOXY(15,12);
  WRITELN (' End of File Encountered !');
  GOTO 11;
END
ELSE BEGIN
  WHILE NOT EOF (INVIN) DO
  BEGIN
    IF RECNO (>) 1 THEN SEEK (INVIN,RECNO-1)
      ELSE RESET (INVIN);
    READ (INVIN,INREC);
    SCREEN_APPEND;
    GOTOXY (15,10); WRITE(RECNO);
    GOTOXY (17,12); WRITE(INREC.CODE);
    GOTOXY (17,13); WRITE(INREC.REPLCOST);
    GOTOXY (17,14); WRITE(INREC.LEADTIME);
    GOTOXY (17,15); WRITE(INREC.LEADVARIANCE);
    SEEK (INVIN,RECNO-1);
    GOTOXY (17,12);
    WHILE NOT KEYSTAT(CH) DO BEGIN END;
    IF NOT (ORD(CH)=17)
      THEN BEGIN
        WRITE (CH);
        GOTOXY (18,12);

```

Line 189 Col 20 Insert Indent A:INVIN.PAS

```

  READ (CV[RECNO]); CLREOL;
  CV[RECNO] := CONCAT(CH,CV[RECNO]);
  GOTOXY (17,13);
  READ (RCV[RECNO]); CLREOL;
  GOTOXY (17,14);
  READ (LTV[RECNO]); CLREOL;
  GOTOXY (17,15);
  READ (LVV[RECNO]); CLREOL;
  1 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
  IF NOT (CH IN COMMANDSET)
    THEN BEGIN
      BEEP; GOTO 1;
    END
  ELSE CASE ORD(CH) OF
    59 : BEGIN
      WITH INREC DO
      BEGIN
        REC_NO      := RECNO;
        CODE        := CV[RECNO];
        REPLCOST    := RCV[RECNO]
        LEADTIME    := LTV[RECNO]
        LEADVARIANCE := LVV[RECNO]
      END;
      WRITE(INVIN,INREC);

```

Line 212 Col 20 Insert Indent A:INVIN.PAS

```

                SCREEN_APPEND;
                END;
            27 : BEGIN
                SCREEN_APPEND;
                RECNO := RECNO-1;
                END;
            17 : SEEK (INVIN,FILESIZE(INVIN))
                END;
        END
    ELSE SEEK (INVIN,FILESIZE(INVIN));
    RECNO := RECNO + 1;
    END;
    END;
    END;
    21 : RECNO := 1;
    SCREEN_MAIN;
    RECNO := 1;
    31 : END;

```

```

'A' : BEGIN
    MAXREC := FILESIZE(NAMEIN)-FILESIZE(INVIN);
    RECNO := FILESIZE(INVIN);
    SEEK (INVIN,RECNO);
    FOR I := 1 TO MAXREC DO

```

Line 231 Col 11 Insert Indent A:INVIN.PAS

```

    BEGIN
        WITH INREC DO
            BEGIN
                CODE      := '.....';
                REPLCOST  := 0;
                LEADTIME  := 0;
                LEADVARIANCE := 0;
            END;
            WRITE (INVIN,INREC);
        END;
        SCREEN_MAIN;
        GOTOXY (15,12);
        WRITELN (' ' ,(MAXREC), ' Records Added');
    END;

```

```

'S' : BEGIN
    RESET (NAMEIN);
    RESET (INVIN);
    I := 1;
    J := 1;
    WHILE NOT EOF (INVIN) DO
        BEGIN
            READ (INVIN,INREC);
            WITH INREC DO

```

Line 254 Col 11 Insert Indent A:INVIN.PAS

```

BEGIN
  RV[I] := REC_NO;
  CV[I] := CODE;
  RCV[I] := REPLCOST;
  LTV[I] := LEADTIME;
  LVV[I] := LEADVARIANCE;
END;
I := I+1;
END;
MAXREC := FILESIZE(INVIN);
REWRITE(INVIN);
WHILE NOT EOF (NAMEIN) DO
BEGIN
  READ (NAMEIN,NAMEREC);
  RECNO := NAMEREC.REC_NO;
  COD := NAMEREC.CODE;
  FOR I := 1 TO MAXREC DO
  BEGIN
    IF COD = CV[I]
      THEN BEGIN
        WITH INREC DO
          BEGIN
            REC_NO      := RECNO;
            CODE        := CV[I];

```

Line 278 Col 1 Insert Indent A:INVIN.PAS

```

  REPLCOST := RCV[I];
  LEADTIME := LTV[I];
  LEADVARIANCE := LVV[I];
END;
WRITE (INVIN,INREC);
END;

```

```

END;
J := J+1;

```

```

END;

```

```

END;

```

```

END;

```

```

END;

```

```

BEGIN

```

```

  SCREEN_MAIN;
  ASSIGN (NAMEIN,'A:NAME.DAT');
  ASSIGN (INVIN,'A:INVIN.DAT');
  RESET(NAMEIN);
  QUIT := FALSE;
  REPEAT
    WHILE NOT KEYSTAT(CH) DO BEGIN END;
    IF CH IN COMMANDSET
      THEN CASE ORD(CH) OF

```

Line 301 Col 1 Insert Indent A:INVIN.PAS

```

1 : BEGIN
  RESET (INVIN);
  BRANCH := 'A';
  PROC_EDIT(NAMEREC,NAMEIN,INREC,INVIN,BRANCH);
END;
3 : BEGIN
  REWRITE (INVIN);
  BRANCH := 'A';
  PROC_EDIT(NAMEREC,NAMEIN,INREC,INVIN,BRANCH);
END;
5 : BEGIN
  BRANCH := 'E';
  RESET (INVIN);
  PROC_EDIT(NAMEREC,NAMEIN,INREC,INVIN,BRANCH);
END;
12 : BEGIN
  RESET(INVIN);
  RECNO := 1;
  SCREEN MAIN;
  GOTOXY(1,10);
  WRITELN (' :5,'NO',' :5,'CODE',' :5,
'REPLENISHMENT COST',' :5,'LEAD TIME',
' :5,'LEAD VARIANCE');
  WHILE NOT EOF(INVIN) DO

```

Line 324 Col 1 Insert Indent A:INVIN.PAS

```

BEGIN
  READ(INVIN,INREC);
  WITH INREC DO BEGIN
    WRITELN(REC_NO:7,CODE:9,REPLCOST:19:2,
      LEADTIME:16,LEADVARIANCE:16:2);END;
  RECNO := RECNO+1;
  END;
  WRITELN;
  WRITE('INVIN File :20,'Number of Records :':30,
    FILESIZE(INVIN));
  END;
16 : BEGIN
  GOTOXY(15,12);WRITELN(') Printing ...');CLREOL;
  RESET(INVIN);
  RECNO := 1;
  WRITELN(LST,'NO':15,'CODE':10,' :5,
'REPLENISHMENT COST':20,'LEAD TIME':20,
'LEAD VARIANCE':20);
  WRITELN(LST,'---':15,'----':10,' :5,
'-----':20,'-----':20,
'-----':20);WRITELN(LST);

```

```

WHILE NOT EOF(INVIN) DO
  BEGIN
    READ(INVIN,INREC);

```

```

Line 347 Col 1 Insert Indent A:INVIN.PAS
  WITH INREC DO BEGIN
    WRITELN(LST,REC_NO:15,CODE:10,REPLCOST:22:3,
      LEADTIME:19,LEADVARIANCE:20:2);END;
    WRITELN(LST);
    RECNO := RECNO+1;
  END;
  WRITELN(LST);WRITELN(LST,' ':15,'INVIN File',
    'Number of Records : ':30,FILESIZE(INVIN));
  GOTOXY(15,12);WRITELN('>> Press ALT-PrtSc');
  END;
17 : QUIT := TRUE;
19 : BEGIN
  BRANCH := 'S';
  PROC_EDIT (NAMEREC,NAMEIN,INREC,INVIN,BRANCH);
  END;
  END
  ELSE BEGIN;
    BEEP;
  END;
  UNTIL QUIT;
CLOSE (INVIN);
CLOSE (NAMEIN);
END.

```

Logged drive: A
Active directory: \

Work file: A:INVIN.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 13467 bytes
Free: 49168 bytes

)

Compiling
359 lines

Code: 011D paragraphs (4560 bytes), 0C60 paragraphs free
Data: 000B paragraphs (176 bytes), 0FDC paragraphs free
Stack/Heap: 2FAF paragraphs (195312 bytes)

)

```
Line 1 Col 1 Insert Indent A:BOMFILE.PAS
```

```
PROGRAM BOMDATA;
{$V-}
TYPE
  STRING4 = STRING[4];
  INPUTRECORD = RECORD
    REC_NO      : INTEGER;
    FATHERCODE  : STRING4;
    SONCODE     : STRING4;
    GOESINTO    : REAL;
  END;
  RECFILE = FILE OF INPUTRECORD;
VAR
  RECNO : INTEGER;
  CH    : CHAR;
  BRANCH: CHAR;
  QUIT  : BOOLEAN;
  INREC : INPUTRECORD;
  BOMFILE : FILE OF INPUTRECORD;
CONST
  COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];
```

```
FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;
```

```
TYPE REG_PACK = RECORD
```

```
Line 25 Col 1 Insert Indent A:BOMFILE.PAS
```

```
  AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
```

```
  END;
```

```
VAR REGISTERS : REG_PACK;
```

```
BEGIN
```

```
  REGISTERS.AX := $0600;
```

```
  REGISTERS.DX := 255;
```

```
  MSDOS(REGISTERS);
```

```
  CH :=CHR(REGISTERS.AX);
```

```
  KEYSTAT := CH (<) CHR(0)
```

```
END;
```

```
PROCEDURE BEEP;
```

```
BEGIN
```

```
  WRITE(CHR(7));
```

```
END;
```

```
PROCEDURE SCREEN_MAIN;
```

```
BEGIN
```

```
  CLRSCR;
```

```
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');
```

```
  GOTOXY (12,2); WRITELN ('! CTRL-C : CLEAR !');
```

```
  GOTOXY (12,3); WRITELN ('! CTRL-A : APPEND !');
```

```

Line 49 Col 1 Insert Indent A:BOMFILE.PAS
GOTOXY (12,4); WRITELN ('! CTRL-E : EDIT      !');
GOTOXY (12,5); WRITELN ('! CTRL-Q : QUIT      !');
GOTOXY (12,6); WRITELN ('! CTRL-L : LIST      !');
GOTOXY (12,7); WRITELN ('! CTRL-P : PRINT     !');
END;

```

```

PROCEDURE SCREEN_APPEND;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');
  GOTOXY (12,2); WRITELN ('! F1   : SAVE      !');
  GOTOXY (12,3); WRITELN ('! ESC  : ESCAPE    !');
  GOTOXY (12,4); WRITELN ('! CTRL-Q : QUIT    !');
  GOTOXY (5,10);
  WRITELN('REC_NO :');WRITELN;
  WRITELN('FATHERCODE :');
  WRITELN('SONCODE   :');
  WRITELN('GOESINTO  :')
END;

```

```

Line 73 Col 1 Insert Indent A:BOMFILE.PAS
PROCEDURE PROC_EDIT (VAR BOMFILE: RECFILE;
                     VAR INREC  : INPUTRECORD;
                     VAR BRANCH : CHAR);
LABEL 1,11,21,31,41,27;
TYPE
  INDEX = 1..100;
  STRING4 = STRING[4];
  RECNOVECTOR = ARRAY [INDEX] OF INTEGER;
  FATHERCODEVECTOR = ARRAY [INDEX] OF STRING4;
  SONCODEVECTOR = ARRAY [INDEX] OF STRING4;
  GOESINTOVECTOR = ARRAY [INDEX] OF REAL;

VAR
  RV : RECNOVECTOR;
  FCV : FATHERCODEVECTOR;
  SCV : SONCODEVECTOR;
  GIV : GOESINTOVECTOR;
  CH : CHAR;
  C : STRING4;
  RECNO : INTEGER;
  EXIT : BOOLEAN;
  ERROR : INTEGER;

CONST
  COMMANDSET : SET OF CHAR = [#17,#27,#59];

```

Line 97 Col 1 Insert Indent A:BOMFILE.PAS
 BEGIN

```

CASE BRANCH OF
'E' :BEGIN
  EXIT := FALSE;
  CLRSCR;
  GOTOXY(10,10);
  WRITELN('FIND RECNO = ');
  I1 : GOTOXY(23,10);
  WHILE NOT KEYSTAT(CH) DO BEGIN END;
  IF ((ORD(CH)=27) OR (ORD(CH)=17))
  THEN GOTO 21
  ELSE BEGIN
    WRITE (CH);
    READ(C);
    C := CONCAT (CH,C);
    VAL (C,RECNO,ERROR);
  END;
  IF FILESIZE(BOMFILE) = 0
  THEN BEGIN
    SCREEN_MAIN;
    GOTOXY (15,12); WRITELN ('File is Empty !');
    GOTO 31;
  END
  ELSE BEGIN

```

Line 121 Col 1 Insert Indent A:BOMFILE.PAS

```

IF RECNO > FILESIZE (BOMFILE)
  THEN BEGIN
    GOTOXY(15,12);
    WRITELN (' End of File Encountered !');
    GOTO 11;
  END
  ELSE BEGIN
    WHILE NOT EOF (BOMFILE) DO
    BEGIN
      IF RECNO < 1 THEN SEEK (BOMFILE,RECNO-1)
      ELSE RESET (BOMFILE);
      READ (BOMFILE,INREC);
      SCREEN_APPEND;
      GOTOXY (15,10); WRITE(RECNO);
      GOTOXY (15,12); WRITE(INREC.FATHERCODE);
      GOTOXY (15,13); WRITE(INREC.SONCODE);
      GOTOXY (15,14); WRITE(INREC.GOESINTO);
      SEEK (BOMFILE,RECNO-1);
      GOTOXY (15,12);
      WHILE NOT KEYSTAT(CH) DO BEGIN END;
      IF NOT (ORD(CH)=17)
      THEN BEGIN
        WRITE (CH);
        GOTOXY (16,12);

```



```

Line 145 Col 1 Insert Indent A:BOMFILE.PAS
      READ (FCV[RECNO]); CLREOL;
      FCV[RECNO] := CONCAT(CH,FCV[RECNO]);
      GOTOXY (15,13);
      READ (SCV[RECNO]); CLREOL;
      GOTOXY (15,14);
      READ (GIV[RECNO]); CLREOL;
      1 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
      IF NOT (CH IN COMMANDSET)
      THEN BEGIN
          BEEP; GOTO 1;
        END
      ELSE CASE ORD(CH) OF
          59 : BEGIN
              WITH INREC DO
              BEGIN
                  REC_NO      := RECNO;
                  FATHERCODE := FCV[RECNO];
                  SONCODE     := SCV[RECNO];
                  GOESINTO    := GIV[RECNO];
              END;
              WRITE(BOMFILE,INREC);
              SCREEN_APPEND;
            END;
          27 : BEGIN

```

```

Line 169 Col 1 Insert Indent A:BOMFILE.PAS
      SCREEN_APPEND;
      RECNO := RECNO-1;
      END;
      17 : SEEK (BOMFILE,
                FILESIZE(BOMFILE));
      END;
      END
    ELSE SEEK (BOMFILE,FILESIZE(BOMFILE));

```

```

      RECNO := RECNO + 1;
      END;
      END;
      END;
      21 : RECNO := 1;
      SCREEN_MAIN;
      31 : END;

```

```

'A' : BEGIN
      RECNO := FILESIZE(BOMFILE);
      SEEK (BOMFILE,RECNO);
      SCREEN_APPEND;
      EXIT := FALSE;
      REPEAT
          27 :GOTOXY(15,10);
          WRITE(RECNO +1);

```

Line 193 Col 1 Insert Indent A:BOMFILE.PAS

```

GOTOXY(15,12);
WHILE NOT KEYSTAT(CH) DO BEGIN END;
IF NOT (ORD(CH) = 17)
  THEN BEGIN
    RECNO := RECNO + 1;
    GOTOXY(15,12);
    WRITE(CH);
    READ(C);
    FCV[RECNO] := CONCAT (CH,C);
    GOTOXY (15,13);
    READ (SCV[RECNO]);
    GOTOXY (15,14);
    READ (GIV[RECNO]);
    41 : WHILE NOT KEYSTAT (CH) DO BEGIN END;
    IF NOT (CH IN COMMANDSET)
      THEN BEGIN
        BEEP;
        GOTO 41;
      END
    ELSE CASE ORD(CH) OF
      59 : BEGIN
        WITH INREC DO
          BEGIN
            REC_NO      := RECNO;

```

Line 217 Col 1 Insert Indent A:BOMFILE.PAS

```

    FATHERCODE := FCV[RECNO];
    SONCODE    := SCV[RECNO];
    GOESINTO   := GIV[RECNO];
    END;
    WRITE(BOMFILE,INREC);
    SCREEN_APPEND;
    END;
    27 : BEGIN
      SCREEN_APPEND;
      RECNO := RECNO-1;
      GOTO 27;
    END;
    17 : EXIT := TRUE;
    END;

```

END

ELSE EXIT := TRUE;

UNTIL EXIT ;

RECNO := 1;

SCREEN_MAIN;

END;

END;

END;

BEGIN

Line 240 Col 1 Insert Indent A:BOMFILE.PAS

```
BEGIN
  SCREEN_MAIN;
  ASSIGN (BOMFILE, 'A:BOMFILE.DAT');
  QUIT := FALSE;
  REPEAT
    WHILE NOT KEYSTAT(CH) DO BEGIN END;
    IF CH IN COMMANDSET
      THEN CASE ORD(CH) OF
        1 : BEGIN
          BRANCH := 'A';
          RESET(BOMFILE);
          PROC_EDIT(BOMFILE, INREC, BRANCH);
        END;
        3 : BEGIN
          BRANCH := 'A';
          REWRITE (BOMFILE);
          PROC_EDIT(BOMFILE, INREC, BRANCH);
        END;
        5 : BEGIN
          BRANCH := 'E';
          RESET (BOMFILE);
          PROC_EDIT(BOMFILE, INREC, BRANCH);
        END;
        12 : BEGIN
```

Line 275 Col 1 Insert Indent A:BOMFILE.PAS

```
  RESET(BOMFILE);
  RECNO := 1;
  GOTOXY(5,10);
  WRITELN(' ':15, 'BOMFILE File');
  WRITELN;
  WRITELN ('REC_NO', ' ':5, 'FATHERCODE',
    ' ':10, 'SONCODE', ' ':10, 'GOESINTO');
  WHILE NOT EOF(BOMFILE) DO
    BEGIN
      READ(BOMFILE, INREC);
      WITH INREC DO BEGIN
        WRITELN(REC_NO:5, FATHERCODE:13,
          SONCODE:18, GOESINTO:19:3);END;
      RECNO := RECNO+1;
    END; WRITELN;
  WRITELN(' ':15, 'No. of Records : ', FILESIZE(BOMF
  END;
  16 : BEGIN
    GOTOXY(10,10); WRITELN(') Printing ...'); CLR
    RESET(BOMFILE);
    RECNO := 1;
    WRITELN(LST, 'NO':15, 'FATHERCODE':20,
      'SONCODE':20, 'GOESINTO':20);
    WRITELN(LST, '--':15, '-----':20,
```

```

Line 298 Col 1 Insert Indent A:BOMFILE.PAS
      '-----':20, '-----':20);WRITELN(LST);
      WHILE NOT EOF(BOMFILE) DO
      BEGIN
        READ(BOMFILE,INREC);
        WITH INREC DO BEGIN
          WRITELN(LST,REC_NO:15,FATHERCODE:17,
            SONCODE:22,GOESINTO:20:3);END;
          WRITELN(LST);
          RECNO := RECNO+1;
        END;
        WRITELN(LST);WRITELN(LST,' ':15,'BOMFILE Fil
          'No. of Records : ':30,FILESIZE(BOMFILE));
        SCREEN_MAIN;
        GOTOXY(10,10);WRITELN(') Press ALT-PrtSC');
      END;
      17 : QUIT := TRUE;
    END
  ELSE BEGIN;
    BEEP;
  END;
UNTIL QUIT;
CLOSE (BOMFILE);
END.

```

Logged drive: A
Active directory: \

Work file: A:BONFILE.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 12334 bytes
Free: 50301 bytes

)

Compiling
 310 lines

Code: 00F6 paragraphs (3936 bytes), 0C87 paragraphs free
Data: 0005 paragraphs (80 bytes), 0FE2 paragraphs free
Stack/Heap: 3022 paragraphs (197152 bytes)

)

Line 1 Col 1 Insert Indent A:DFILE.PAS

PROGRAM DEMANDDATA;

TYPE

```

STRING4 = STRING[4];
STRING20 = STRING[20];
NAMERECORD = RECORD
    REC_NO : INTEGER;
    CODE : STRING4;
    NAME : STRING20;
    UNIT : STRING4;
    CLASS : STRING4;
    NAMELENGTH : INTEGER;
END;
PERNAMERECORD = RECORD
    REC_NO : INTEGER;
    PERNAME : STRING20;
    NAMELENGTH : INTEGER;
END;
NAMEFILE = FILE OF NAMERECORD;
PERNAMEFILE = FILE OF PERNAMERECORD;
RECFILE = FILE OF REAL;

```

VAR

```

CH : CHAR;
I : INTEGER;
BRANCH : CHAR;

```

Line 35 Col 1 Insert Indent A:DFILE.PAS

```

DEMAND : REAL;
QUIT : BOOLEAN;
NAMEREC : NAMERECORD;
PERNAMEREC : PERNAMERECORD;
NAMEIN : FILE OF NAMERECORD;
PERNAMEIN : FILE OF PERNAMERECORD;
DFILE : FILE OF REAL;

```

CONST

```

COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];

```

FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;

TYPE REG_PACK = RECORD

```

    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
END;

```

VAR REGISTERS : REG_PACK;

BEGIN

```

REGISTERS.AX := $0600;
REGISTERS.DX := 255;
MSDOS(REGISTERS);
CH := CHR(REGISTERS.AX);
KEYSTAT := CH <> CHR(0)

```

END;

Line 53 Col 1 Insert Indent A:DFILE.PAS

```
PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;
```

```
PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! ***** MENU **** !');
  GOTOXY (12,2); WRITELN ('! CTRL-C : CLEAR !');
  GOTOXY (12,3); WRITELN ('! CTRL-A : APPEND !');
  GOTOXY (12,4); WRITELN ('! CTRL-E : EDIT !');
  GOTOXY (12,5); WRITELN ('! CTRL-P : PRINT !');
  GOTOXY (12,6); WRITELN ('! CTRL-Q : QUIT !');
END;
```

```
PROCEDURE SCREEN_EDIT;
```

Line 76 Col 58 Insert Indent A:DFILE.PAS

```
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! ***** MENU **** !');
  GOTOXY (12,2); WRITELN ('! CTRL-E : EDIT !');
  GOTOXY (12,3); WRITELN ('! CTRL-S : SKIP PAGE !');
  GOTOXY (12,4); WRITELN ('! CTRL-Q : QUIT !');
  GOTOXY (5,7); WRITELN ('PERIOD : ');
  GOTOXY (5,9); WRITELN ('NO', ' ':5,'CODE', ' ':9,'DEMAND');
END;
```

```
PROCEDURE PROC_EDIT (VAR DFILE : RECFILE;
  VAR BRANCH : CHAR;
  VAR NAMEIN : NAMEFILE;
  VAR NAMEREC : NAMERECORD;
  VAR PERNAMEIN : PERNAMEFILE;
  VAR PERNAMEREC : PERNAMERECORD);
```

```
TYPE
  INDEX = 1..120;
```

```

Line 116 Col 58 Insert Indent A:DFILE.PAS
STRING4 = STRING[4];
STRING20 = STRING[20];
CODEVECTOR = ARRAY [INDEX] OF STRING4;
PERNAMEVECTOR = ARRAY [INDEX] OF STRING20;
DEMANDVECTOR = ARRAY [INDEX] OF REAL;
DEMANDMATRIX = ARRAY [1..20,1..20] OF REAL;

```

```
VAR
```

```

CV : CODEVECTOR;
PNV : PERNAMEVECTOR;
DV : DEMANDVECTOR;
D : DEMANDMATRIX;
CH : CHAR;
I,PI : INDEX;
NL : INTEGER;
RECNO : INTEGER;
MAXREC : INTEGER;
MAXITEM : INTEGER;
MAXPERIOD : INTEGER;
EXIT : BOOLEAN;

```

```
CONST
```

```
COMMANDSET : SET OF CHAR = [#17,#27,#19,#5];
```

```
BEGIN
```

```

MAXITEM := FILESIZE(NAMEIN);
MAXPERIOD := FILESIZE(PERNAMEIN);

```

```
Line 139 Col 58 Insert Indent A:DFILE.PAS
```

```
MAXREC := MAXITEM*MAXPERIOD;
```

```
CASE BRANCH OF
```

```
'E' :BEGIN
```

```
  I := 1;
```

```
  WHILE NOT EOF (DFILE) DO
```

```
  BEGIN
```

```
    READ (DFILE,DEMAND);
```

```
    DV[I] := DEMAND;
```

```
    I := I+1;
```

```
  END;
```

```
  RESET (DFILE);
```

```
  FOR I := 1 TO MAXITEM DO
```

```
  BEGIN
```

```
    READ (NAMEIN,NAMEREC);
```

```
    CV[I] := NAMEREC.CODE;
```

```
  END;
```

```
  PI := 1;
```

```
  WHILE NOT EOF (PERNAMEIN) DO
```

```
  BEGIN
```

```
    SCREEN_EDIT;
```

```
    READ (PERNAMEIN,PERNAMEREC);
```

```
    PNV[PI] := PERNAMEREC.PERNAME;
```

```
    GOTOXY (14,7);
```

```
    WRITELN (PNV[PI]);
```


Line 162 Col 58 Insert Indent A:DFILE.PAS

```

GOTOXY (1,11);
FOR I := 1 TO MAXITEM DO
BEGIN
  RECNO := (PI-1)*MAXITEM + I;
  WRITELN (RECNO:6,CV[I]:9);
  GOTOXY (WHEREX+20,WHEREY-1);
  WRITELN (DV[RECNO]:12:3);
END;
GOTOXY (25,11);
REPEAT
  WHILE NOT KEYSTAT (CH) DO BEGIN END;
  IF NOT (CH IN COMMANDSET) THEN BEEP;
UNTIL (CH IN COMMANDSET);
  CASE ORD(CH) OF
    5 : BEGIN
      FOR I := 1 TO MAXITEM DO
      BEGIN
        READ (DEMAND);
        RECNO := (PI-1)*MAXITEM+I;
        DV [RECNO] := DEMAND;
        GOTOXY (25,WHEREY+1);
      END;
    END;
  17 : SEEK (PERNAMEIN,MAXPERIOD);

```

Line 185 Col 58 Insert Indent A:DFILE.PAS

```

  END;
  PI := PI+1;
END;
SCREEN_MAIN;
RESET (DFILE);
PI := 1;
WHILE NOT EOF(DFILE) DO
BEGIN
  DEMAND := DV[PI];
  WRITE(DFILE,DEMAND);
  PI := PI+1;
END;
END;

```

'A' : BEGIN

```

MAXREC := FILESIZE(NAMEIN)*FILESIZE(PERNAMEIN)-FILESIZE(DFILE);
RECNO := FILESIZE(DFILE);
SEEK (DFILE,RECNO);
FOR I := 1 TO MAXREC DO
BEGIN
  DV[I] := 0;
  DEMAND :=0;
  WRITE (DFILE,DEMAND);
END;

```

Line 208 Col 58 Insert Indent A:DFILE.PAS

```
SCREEN_MAIN;
GOTOXY (15,12);
WRITELN (' ' ,(MAXREC), ' Records Added');
END;
```

'P' : BEGIN

```
SCREEN_MAIN;GOTOXY(10,10);WRITELN('>> Printing ... ');
WRITELN(LST,' ':40,'TIME-PHASDE DEMAND');WRITELN(LST);
WRITELN(LST);WRITE(LST,'NAME':10,' ':17);
FOR PI := 1 TO MAXPERIOD DO
BEGIN
  READ(PERNAMEIN,PERNAMEREC);
  WRITE(LST,PERNAMEREC.PERNAME:12);
  FOR I := 1 TO MAXITEM DO
    READ (DFILE,D[I,PI]);
  END;
  WRITE(LST);WRITE(LST,'----':10,' ':17);
  FOR PI := 1 TO MAXPERIOD DO WRITE(LST,'-----':12);WRITELN(LST);
  FOR I := 1 TO MAXITEM DO
  BEGIN
    READ(NAMEIN,NAMEREC);
    NL := LENGTH(NAMEREC.NAME);
    WRITE(LST);WRITELN(LST);WRITE(LST,' ':6,NAMEREC.NAME:NL,' ':21-
    FOR PI := 1 TO MAXPERIOD DO
```

Line 223 Col 1 Insert Indent A:DFILE.PAS

```
WRITE(LST,D[I,PI]:12:3);
```

```
END;
```

```
WRITELN(LST);SCREEN_MAIN;
```

```
END;
```

```
END;
```

```
END;
```

BEGIN

```
SCREEN_MAIN;
ASSIGN (NAMEIN,'A:NAME.DAT');
ASSIGN (PERNAMEIN,'A:PERNAME.DAT');
ASSIGN (DFILE,'A:DFILE.DAT');
QUIT := FALSE;
REPEAT
  RESET (NAMEIN);
  RESET (PERNAMEIN);
  WHILE NOT KEYSTAT(CH) DO BEGIN END;
  IF CH IN COMMANDSET
    THEN CASE ORD(CH) OF
      1 : BEGIN
          RESET (DFILE);
          BRANCH := 'A';
```

```

Line 246 Col 1 Insert Indent A:DFILE.PAS
      PROC_EDIT(DFILE,BRANCH,NAMEIN,NAMEREC,
                PERNAMEIN,PERNAMEREC);
      END;
3 : BEGIN
      REWRITE (DFILE);
      BRANCH := 'A';
      PROC_EDIT(DFILE,BRANCH,NAMEIN,NAMEREC,
                PERNAMEIN,PERNAMEREC);
      END;
5 : BEGIN
      RESET (DFILE);
      BRANCH := 'E';
      PROC_EDIT(DFILE,BRANCH,NAMEIN,NAMEREC,
                PERNAMEIN,PERNAMEREC);
      END;
12 : BEGIN
      RESET (DFILE);
      I := 1;
      WHILE NOT EOF(DFILE) DO
      BEGIN
        READ(DFILE,DEMAND);
        WRITELN(LST,I,DEMAND);
        I := I+1;
      END;

```

```

Line 284 Col 1 Insert Indent A:DFILE.PAS
      END;
16 : BEGIN
      RESET (DFILE);
      BRANCH := 'P';
      PROC_EDIT(DFILE,BRANCH,NAMEIN,NAMEREC,
                PERNAMEIN,PERNAMEREC);
      END;
17 : QUIT := TRUE;
      END
      ELSE BEEP;
      UNTIL QUIT;
CLOSE (NAMEIN);
CLOSE (PERNAMEIN);
CLOSE (DFILE);
END.

```

Logged drive: A
Active directory: \

Work file: A:D\FILE.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 8836 bytes
Free: 53799 bytes

)

Compiling
 276 lines

Code: 00D0 paragraphs (3328 bytes), 0CAD paragraphs free
Data: 000E paragraphs (224 bytes), 0FD9 paragraphs free
Stack/Heap: 311A paragraphs (201120 bytes)

)

Line 1 Col 1 Insert Indent A:PFILE.PAS

```
PROGRAM PRODUCTIONDATA;
```

```
TYPE
```

```
  STRING4 = STRING[4];
```

```
  STRING20 = STRING[20];
```

```
  NAMERECORD = RECORD
```

```
    REC_NO : INTEGER;
```

```
    CODE : STRING4;
```

```
    NAME : STRING20;
```

```
    UNIT : STRING4;
```

```
    CLASS : STRING4;
```

```
    NAMELENGTH : INTEGER;
```

```
  END;
```

```
  PERNAMERECORD = RECORD
```

```
    REC_NO : INTEGER;
```

```
    PERNAME : STRING20;
```

```
    NAMELENGTH : INTEGER;
```

```
  END;
```

```
  NAMEFILE = FILE OF NAMERECORD;
```

```
  PERNAMEFILE = FILE OF PERNAMERECORD;
```

```
  RECFILE = FILE OF REAL;
```

```
VAR
```

```
  CH : CHAR;
```

```
  I : INTEGER;
```

```
  PRODUCTION : REAL;
```

Line 35 Col 1 Insert Indent A:PFILE.PAS

```
BRANCH : CHAR;
```

```
QUIT : BOOLEAN;
```

```
NAMEREC : NAMERECORD;
```

```
PERNAMEREC : PERNAMERECORD;
```

```
NAMEIN : FILE OF NAMERECORD;
```

```
PERNAMEIN : FILE OF PERNAMERECORD;
```

```
PFILE : FILE OF REAL;
```

```
CONST
```

```
  COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];
```

```
FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;
```

```
TYPE REG_PACK = RECORD
```

```
  AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
```

```
END;
```

```
VAR REGISTERS : REG_PACK;
```

```
BEGIN
```

```
  REGISTERS.AX := $0600;
```

```
  REGISTERS.DX := 255;
```

```
  MSDOS(REGISTERS);
```

```
  CH :=CHR(REGISTERS.AX);
```

```
  KEYSTAT := CH (<) CHR(0)
```

```
END;
```

Line 49 Col 1 Insert Indent A:PFILE.PAS

```
PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;
```

```
PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('| ***** MENU ***** |');
  GOTOXY (12,2); WRITELN ('| CTRL-C : CLEAR      |');
  GOTOXY (12,3); WRITELN ('| CTRL-A : APPEND    |');
  GOTOXY (12,4); WRITELN ('| CTRL-E : EDIT      |');
  GOTOXY (12,5); WRITELN ('| CTRL-P : PRINT     |');
  GOTOXY (12,6); WRITELN ('| CTRL-Q : QUIT      |');
END;
```

Line 73 Col 1 Insert Indent A:PFILE.PAS

```
PROCEDURE SCREEN_EDIT;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('| ***** MENU ***** |');
  GOTOXY (12,2); WRITELN ('| CTRL-E : EDIT      |');
  GOTOXY (12,3); WRITELN ('| CTRL-S : SKIP PAGE |');
  GOTOXY (12,4); WRITELN ('| CTRL-Q : QUIT      |');
  GOTOXY (5,7); WRITELN ('PERIOD : ');
  GOTOXY (5,9); WRITELN ('NO', ' ':5, 'CODE', ' ':9, 'PRODUCTION');
END;
```

```
PROCEDURE PROC_EDIT (VAR PFILE : RECFILE;
  VAR BRANCH : CHAR;
  VAR NAMEIN : NAMEFILE;
  VAR NAMEREC : NAMERECORD;
  VAR PERNAMEIN : PERNAMEFILE;
  VAR PERNAMEREC : PERNAMERECORD);
```

```
TYPE
  INDEX = 1..120;
  STRING4 = STRING[4];
  STRING20 = STRING[20];
```

```

Line 97 Col 1 Insert Indent A:PFILE.PAS
CODEVECTOR      = ARRAY [INDEX] OF STRING4;
PERNAMEVECTOR   = ARRAY [INDEX] OF STRING20;
PRODUCTIONVECTOR = ARRAY [INDEX] OF REAL;
PRODUCTIONMATRIX = ARRAY [1..20,1..20] OF REAL;
VAR
CV : CODEVECTOR;
PNV : PERNAMEVECTOR;
PRDV : PRODUCTIONVECTOR;
P : PRODUCTIONMATRIX;
PRODUCTION : REAL;
CH : CHAR;
I,PI : INDEX;
NL : INTEGER;
RECNO : INTEGER;
MAXREC : INTEGER;
MAXITEM : INTEGER;
MAXPERIOD : INTEGER;
EXIT : BOOLEAN;
CONST
COMMANDSET : SET OF CHAR = [#17,#27,#19,#5];
BEGIN
MAXITEM := FILESIZE(NAMEIN);
MAXPERIOD := FILESIZE(PERNAMEIN);
MAXREC := MAXITEM*MAXPERIOD;

```

```

Line 121 Col 1 Insert Indent A:PFILE.PAS
CASE BRANCH OF
'E' :BEGIN
I := 1;
WHILE NOT EOF (PFILE) DO
BEGIN
READ (PFILE,PRODUCTION);
PRDV[I] := PRODUCTION;
I := I+1;
END;
RESET (PFILE);
FOR I := 1 TO MAXITEM DO
BEGIN
READ (NAMEIN,NAMEREC);
CV[I] := NAMEREC.CODE;
END;
PI := 1;
WHILE NOT EOF (PERNAMEIN) DO
BEGIN
SCREEN_EDIT;
READ (PERNAMEIN,PERNAMEREC);
PNV[PI] := PERNAMEREC.PERNAME;
GOTOXY (14,7);
WRITELN (PNV[PI]);
GOTOXY (1,11);

```

Line 145 Col 1 Insert Indent A:PFILE.PAS

```

FOR I := 1 TO MAXITEM DO
  BEGIN
    RECNO := (PI-1)*MAXITEM + I;
    WRITELN (RECNO:6,CV[1]:9);
    GOTOXY (WHEREX+20,WHEREY-1);
    WRITELN (PRDV[RECNO]:12:3);
  END;
  GOTOXY (25,11);
  REPEAT
    WHILE NOT KEYSTAT (CH) DO BEGIN END;
    IF NOT (CH IN COMMANDSET) THEN BEEP;
  UNTIL (CH IN COMMANDSET);
  CASE ORD(CH) OF
    5 : BEGIN
      FOR I := 1 TO MAXITEM DO
        BEGIN
          READ (PRODUCTION);
          RECNO := (PI-1)*MAXITEM+I;
          PRDV [RECNO] := PRODUCTION;
          GOTOXY (25,WHEREY+1);
        END;
      END;
    17 : SEEK (PERNAMEIN,MAXPERIOD);
  END;

```

Line 169 Col 1 Insert Indent A:PFILE.PAS

```

  PI := PI+1;
END;
SCREEN_MAIN;
RESET (PFILE);
PI := 1;
WHILE NOT EOF(PFILE) DO
  BEGIN
    PRODUCTION := PRDV[PI];
    WRITE(PFILE,PRODUCTION);
    PI := PI+1;
  END;
END;

```

'A' : BEGIN

```

MAXREC := FILESIZE(NAMEIN)*FILESIZE(PERNAMEIN)-FILESIZE(PFILE);
RECNO := FILESIZE(PFILE);
SEEK (PFILE,RECNO);
FOR I := 1 TO MAXREC DO
  BEGIN
    PRDV[I] := 0;
    PRODUCTION :=0;
    WRITE (PFILE,PRODUCTION);
  END;
SCREEN_MAIN;

```



```

Line 193 Col 1 Insert Indent A:PFILE.PAS
  GOTOXY (15,12);
  WRITELN (' ') ,(MAXREC), ' Records Added');
  END;

```

```

'P' : BEGIN
  SCREEN_MAIN;GOTOXY(10,10);WRITELN('') Printing ... ');
  WRITELN(LST,' ':30,'ALREADY SCHEDULED PRODUCTION');WRITELN(LST);
  WRITELN(LST);WRITE(LST,'NAME':10,' ':17);
  FOR PI := 1 TO MAXPERIOD DO
  BEGIN
    READ(PERNAMEIN,PERNAMEREC);
    WRITE(LST,PERNAMEREC.PERNAME:12);
    FOR I := 1 TO MAXITEM DO
      READ (PFILE,P[I,PI]);
    END;
    WRITELN(LST);WRITE(LST,'----':10,' ':17);
    FOR PI := 1 TO MAXPERIOD DO WRITE(LST,'-----':12);WRITELN(LST);
    FOR I := 1 TO MAXITEM DO
      BEGIN
        READ(NAMEIN,NAMEREC);
        NL := LENGTH(NAMEREC.NAME);
        WRITELN(LST);WRITELN(LST);WRITE(LST,' ':6,NAMEREC.NAME:NL,' ':21-NL)
        FOR PI := 1 TO MAXPERIOD DO
          WRITE(LST,P[I,PI]:12:3);

```

```

Line 217 Col 1 Insert Indent A:PFILE.PAS
  END;
  WRITELN(LST);SCREEN_MAIN;
  END;

```

```

END;
END;

```

```

BEGIN
  SCREEN_MAIN;
  ASSIGN (NAMEIN,'A:NAME.DAT');
  ASSIGN (PERNAMEIN,'A:PERNAME.DAT');
  ASSIGN (PFILE,'A:PFILE.DAT');
  QUIT := FALSE;
  REPEAT
    RESET (NAMEIN);
    RESET (PERNAMEIN);
    WHILE NOT KEYSTAT(CH) DO BEGIN END;
    IF CH IN COMMANDSET
      THEN CASE ORD(CH) OF
        1 : BEGIN
          RESET (PFILE);
          BRANCH := 'A';
          PROC_EDIT(PFILE,BRANCH,NAMEIN,NAMEREC,
            PERNAMEIN,PERNAMEREC);
        END;

```

Line 241 Col 1 Insert Indent A:PFILE.PAS

```

3 : BEGIN
    REWRITE (PFILE);
    BRANCH := 'A';
    PROC_EDIT(PFILE,BRANCH,NAMEIN,NAMEREC,
              PERNAMEIN,PERNAMEREC);
    END;
5 : BEGIN
    RESET (PFILE);
    BRANCH := 'E';
    PROC_EDIT(PFILE,BRANCH,NAMEIN,NAMEREC,
              PERNAMEIN,PERNAMEREC);
    END;
12 : BEGIN
    RESET (PFILE);
    I := 1;
    WHILE NOT EOF(PFILE) DO
    BEGIN
        READ(PFILE,PRODUCTION);
        WRITELN(LST,I,PRODUCTION);
        I := I+1;
    END;
    END;
16 : BEGIN
    RESET (PFILE);

```

Line 287 Col 1 Insert Indent A:PFILE.PAS

```

    BRANCH := 'P';
    PROC_EDIT(PFILE,BRANCH,NAMEIN,NAMEREC,
              PERNAMEIN,PERNAMEREC);
    END;
17 : QUIT := TRUE;
    END
    ELSE BEEP;
    UNTIL QUIT;
CLOSE (NAMEIN);
CLOSE (PERNAMEIN);
CLOSE (PFILE);
END.

```

Logged drive: A
Active directory: \

Work file: A:PFILE.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 8911 bytes
Free: 53724 bytes

)

Compiling
 276 lines

Code: 00D1 paragraphs (3344 bytes), 0CAC paragraphs free
Data: 000E paragraphs (224 bytes), 0FD9 paragraphs free
Stack/Heap: 3114 paragraphs (201024 bytes)

)

Line 1 Col 1 Insert Indent A:SFILE.PAS

PROGRAM STORAGEDATA;

TYPE

```

STRING4 = STRING[4];
STRING20 = STRING[20];
NAMERECORD = RECORD
    REC_NO : INTEGER;
    CODE : STRING4;
    NAME : STRING20;
    UNIT : STRING4;
    CLASS : STRING4;
    NAMELENGTH : INTEGER;
END;
PERNAMERECORD = RECORD
    REC_NO : INTEGER;
    PERNAME : STRING20;
    NAMELENGTH : INTEGER;
END;
NAMEFILE = FILE OF NAMERECORD;
PERNAMEFILE = FILE OF PERNAMERECORD;
RECFILE = FILE OF REAL;

```

VAR

```

CH : CHAR;
I : INTEGER;
STORAGE: REAL;

```

Line 35 Col 1 Insert Indent A:SFILE.PAS

```

BRANCH : CHAR;
QUIT : BOOLEAN;
NAMEREC : NAMERECORD;
PERNAMEREC : PERNAMERECORD;
NAMEIN : FILE OF NAMERECORD;
PERNAMEIN : FILE OF PERNAMERECORD;
SFILE : FILE OF REAL;

```

CONST

```

COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];

```

FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;

TYPE REG_PACK = RECORD

```

    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
END;

```

VAR REGISTERS : REG_PACK;

BEGIN

```

    REGISTERS.AX := $0600;
    REGISTERS.DX := 255;
    MSDOS(REGISTERS);
    CH :=CHR(REGISTERS.AX);
    KEYSTAT := CH <> CHR(0)

```

END;

```

Line 70 Col 1 Insert Indent A:SFILE.PAS
PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;

```

```

PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('| ***** MENU ***** |');
  GOTOXY (12,2); WRITELN ('| CTRL-C : CLEAR      |');
  GOTOXY (12,3); WRITELN ('| CTRL-A : APPEND    |');
  GOTOXY (12,4); WRITELN ('| CTRL-E : EDIT      |');
  GOTOXY (12,5); WRITELN ('| CTRL-Q : QUIT      |');
  GOTOXY (12,6); WRITELN ('| CTRL-L : LIST      |');
  GOTOXY (12,7); WRITELN ('| CTRL-P : PRINT     |');
END;

```

```

PROCEDURE SCREEN_EDIT;
BEGIN
  CLRSCR;

Line 82 Col 1 Insert Indent A:SFILE.PAS
  GOTOXY (12,1); WRITELN ('| ***** MENU ***** |');
  GOTOXY (12,2); WRITELN ('| CTRL-E : EDIT      |');
  GOTOXY (12,3); WRITELN ('| CTRL-Q : QUIT      |');
  GOTOXY (5,6); WRITELN ('|NO', ' ':5,'CODE', ' ':9,'STORAGE CAPACITY');
END;

```

```

PROCEDURE PROC_EDIT (VAR SFILE : RECFILE;
                     VAR BRANCH : CHAR;
                     VAR NAMEIN : NAMEFILE;
                     VAR NAMEREC : NAMERECORD;
                     VAR PERNAMEIN : PERNAMEFILE;
                     VAR PERNAMEREC : PERNAMERECORD);

```

```

TYPE
  INDEX      = 1..20;
  STRING4    = STRING[4];
  STRING20   = STRING[20];
  CODEVECTOR = ARRAY [INDEX] OF STRING4;
  PERNAMEVECTOR = ARRAY [INDEX] OF STRING20;
  STORAGEVECTOR = ARRAY [INDEX] OF REAL;

```

```

VAR
  CV : CODEVECTOR;
  PNV : STRING20;

```

Line 105 Col 1 Insert Indent A:SFILE.PAS

```
SV : STORAGEVECTOR;
STORAGE : REAL;
CH : CHAR;
I,PI : INDEX;
RECNO : INTEGER;
MAXREC : INTEGER;
MAXITEM : INTEGER;
MAXPERIOD : INTEGER;
EXIT : BOOLEAN;
```

CONST

```
COMMANDSET : SET OF CHAR = [#17,#27,#5];
```

BEGIN

```
MAXITEM := FILESIZE(NAMEIN);
MAXPERIOD := FILESIZE(PERNAMEIN);
MAXREC := MAXITEM;
```

CASE BRANCH OF

'E' :BEGIN

```
  I := 1;
  WHILE NOT EOF (SFILE) DO
  BEGIN
    READ (SFILE,STORAGE);
    SV[I] := STORAGE;
    I := I+1;
  END;
```

Line 128 Col 1 Insert Indent A:SFILE.PAS

```
RESET (SFILE);
FOR I := 1 TO MAXITEM DO
BEGIN
  READ (NAMEIN,NAMEREC);
  CV[I] := NAMEREC.CODE;
```

END;

SCREEN_EDIT;

```
GOTOXY (1,10);
FOR I := 1 TO MAXITEM DO
BEGIN
  RECNO := I;
  WRITELN (RECNO:6,CV[I]:9);
  GOTOXY (WHEREX+25,WHEREY-1);
  WRITELN (SV[RECNO]:12:3);
```

END;

GOTOXY (30,10);

REPEAT

```
  WHILE NOT KEYSTAT (CH) DO BEGIN END;
  IF NOT (CH IN COMMANDSET) THEN BEEP;
```

UNTIL (CH IN COMMANDSET);

CASE ORD(CH) OF

5 : BEGIN

```
  FOR I := 1 TO MAXITEM DO
  BEGIN
```

Line 151 Col 1 Insert Indent A:SFILE.PAS

```

        READ (STORAGE);
        RECNO := I;
        SV [RECNO] := STORAGE;
        GOTOXY (30,WHEREY+1);
    END;
END;
17 : I := MAXITEM;
END;
SCREEN_MAIN;
RESET (SFILE);
PI := 1;
WHILE NOT EOF(SFILE) DO
BEGIN
    STORAGE := SV[PI];
    WRITE(SFILE,STORAGE);
    PI := PI+1;
END;
END;

```

```

'A' : BEGIN
    MAXREC := FILESIZE(NAMEIN)-FILESIZE(SFILE);
    RECNO := FILESIZE(SFILE);
    SEEK (SFILE,RECNO);
    FOR I := 1 TO MAXREC DO

```

Line 184 Col 63 Insert Indent A:SFILE.PAS

```

BEGIN
    SV[I] := 0;
    STORAGE :=0;
    WRITE (SFILE,STORAGE);
END;
SCREEN_MAIN;
GOTOXY (15,12);
WRITELN (' ' ,(MAXREC), ' Records Added');
END;

```

```

'P' : BEGIN
    SCREEN_MAIN;GOTOXY(10,10);WRITELN(')> Printing ...');
    WRITELN(LST,'CODE':30,' ':10,'NAME','STORAGE CAPACITY':42);
    WRITELN(LST,'----':30,' ':10,'----','-----':42);
    RESET (NAMEIN);
    FOR I := 1 TO MAXITEM DO
    BEGIN
        READ(SFILE,STORAGE);
        READ(NAMEIN,NAMEREC);
        WITH NAMEREC DO
            WRITELN(LST,CODE:30,' ':10,'NAME':NAMELENGTH,' ':20-NAMELENGTH,
                STORAGE:20:3,UNIT:6);WRITELN(LST);
    END;
    SCREEN_MAIN;

```

Line 207 Col 63 Insert Indent A:SFILE.PAS

END;

END;

END;

BEGIN

SCREEN_MAIN;

ASSIGN (NAMEIN, 'A:NAME.DAT');

ASSIGN (PERNAMEIN, 'A:PERNAME.DAT');

ASSIGN (SFILE, 'A:SFILE.DAT');

QUIT := FALSE;

REPEAT

RESET (NAMEIN);

RESET (PERNAMEIN);

WHILE NOT KEYSTAT(CH) DO BEGIN END;

IF CH IN COMMANDSET

THEN CASE ORD(CH) OF

1 : BEGIN

RESET (SFILE);

BRANCH := 'A';

PROC_EDIT(SFILE, BRANCH, NAMEIN, NAMEREC,
PERNAMEIN, PERNAMEREC);

END;

3 : BEGIN

REWRITE (SFILE);

Line 230 Col 63 Insert Indent A:SFILE.PAS

BRANCH := 'A';

PROC_EDIT(SFILE, BRANCH, NAMEIN, NAMEREC,
PERNAMEIN, PERNAMEREC);

END;

5 : BEGIN

RESET (SFILE);

BRANCH := 'E';

PROC_EDIT(SFILE, BRANCH, NAMEIN, NAMEREC,
PERNAMEIN, PERNAMEREC);

END;

12 : BEGIN

RESET (SFILE); WRITELN;

WRITELN(I:10, 'STORAGE CAPACITY':20); WRITELN;

I := 1;

WHILE NOT EOF(SFILE) DO

BEGIN

READ(SFILE, STORAGE);

WRITELN(I:10, STORAGE:20:3);

I := I+1;

END;

WRITELN;

WRITE ('SFILE':20, 'Number of records : ':30,
FILESIZE(SFILE));

END;


```

Line 253 Col 63 Insert Indent A:SFILE.PAS
      16 : BEGIN
            RESET (SFILE);
            BRANCH := 'P';
            PROC_EDIT(SFILE,BRANCH,NAMEIN,NAMEREC,
                    PERNAMEIN,PERNAMEREC);
            END;
      17 : QUIT := TRUE;
    END
  ELSE BEEP;
UNTIL QUIT;
CLOSE (NAMEIN);
CLOSE (PERNAMEIN);
CLOSE (SFILE);
END.

```

Logged drive: A
Active directory: \

Work file: A:SFILE.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 8294 bytes
Free: 54341 bytes

)

Compiling
256 lines

Code: 00B9 paragraphs (2960 bytes), 0CC4 paragraphs free
Data: 000E paragraphs (224 bytes), 0FD9 paragraphs free
Stack/Heap: 3153 paragraphs (202032 bytes)

)

Line 1 Col 1 Insert Indent A:OSFILE.PAS

PROGRAM ONHANDSTOCKDATA;

TYPE

STRING4 = STRING[4];

STRING20 = STRING[20];

NAMERECORD = RECORD

REC_NO : INTEGER;

CODE : STRING4;

NAME : STRING20;

UNIT : STRING4;

CLASS : STRING4;

NAMELENGTH : INTEGER;

END;

PERNAMERECORD = RECORD

REC_NO : INTEGER;

PERNAME : STRING20;

NAMELENGTH : INTEGER;

END;

NAMEFILE = FILE OF NAMERECORD;

PERNAMEFILE = FILE OF PERNAMERECORD;

RECFILE = FILE OF REAL;

VAR

CH : CHAR;

I : INTEGER;

ONHANDSTOCK : REAL;

Line 34 Col 1 Insert Indent A:OSFILE.PAS

ONHANDSTOCK : REAL;

BRANCH : CHAR;

QUIT : BOOLEAN;

NAMEREC : NAMERECORD;

PERNAMEREC : PERNAMERECORD;

NAMEIN : FILE OF NAMERECORD;

PERNAMEIN : FILE OF PERNAMERECORD;

OSFILE : FILE OF REAL;

CONST

COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];

FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;

TYPE REG_PACK = RECORD

AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;

END;

VAR REGISTERS : REG_PACK;

BEGIN

REGISTERS.AX := \$0600;

REGISTERS.DX := 255;

MSDOS(REGISTERS);

CH := CHR(REGISTERS.AX);

KEYSTAT := CH <> CHR(0)

END;

```

Line 69 Col 1 Insert Indent A:OSFILE.PAS
PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;

```

```

PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');
  GOTOXY (12,2); WRITELN ('! CTRL-C : CLEAR !');
  GOTOXY (12,3); WRITELN ('! CTRL-A : APPEND !');
  GOTOXY (12,4); WRITELN ('! CTRL-E : EDIT !');
  GOTOXY (12,5); WRITELN ('! CTRL-Q : QUIT !');
  GOTOXY (12,6); WRITELN ('! CTRL-L : LIST !');
  GOTOXY (12,7); WRITELN ('! CTRL-P : PRINT !');
END;

```

```

PROCEDURE SCREEN_EDIT;
BEGIN
  CLRSCR;

Line 82 Col 1 Insert Indent A:OSFILE.PAS
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');
  GOTOXY (12,2); WRITELN ('! CTRL-E : EDIT !');
  GOTOXY (12,3); WRITELN ('! CTRL-Q : QUIT !');
  GOTOXY (5,6); WRITELN ('NO', ' :5, 'CODE', ' :9, 'ONHAND STOCK');
END;

```

```

PROCEDURE PROC_EDIT (VAR OSFILE : RECFILE;
  VAR BRANCH : CHAR;
  VAR NAMEIN : NAMEFILE;
  VAR NAMEREC : NAMERECORD;
  VAR PERNAMEIN : PERNAMEFILE;
  VAR PERNAMEREC : PERNAMERECORD);

```

```

TYPE
  INDEX = 1..20;
  STRING4 = STRING[4];
  STRING20 = STRING[20];
  CODEVECTOR = ARRAY [INDEX] OF STRING4;
  PERNAMEVECTOR = ARRAY [INDEX] OF STRING20;
  ONHANDSTOCKVECTOR = ARRAY [INDEX] OF REAL;
VAR
  CV : CODEVECTOR;
  PNV : STRING20;

```

```

Line 105 Col 1 Insert Indent A:OSFILE.PAS
OSV : ONHANDSTOCKVECTOR;
ONHANDSTOCK : REAL;
CH : CHAR;
I,PI : INDEX;
RECNO : INTEGER;
MAXREC : INTEGER;
MAXITEM : INTEGER;
MAXPERIOD : INTEGER;
EXIT : BOOLEAN;
CONST
COMMANDSET : SET OF CHAR = [#17,#27,#5];
BEGIN
MAXITEM := FILESIZE(NAMEIN);
MAXPERIOD := FILESIZE(PERNAMEIN);
MAXREC := MAXITEM;
CASE BRANCH OF
'E' :BEGIN
I := 1;
WHILE NOT EOF (OSFILE) DO
BEGIN
READ (OSFILE,ONHANDSTOCK);
OSV[I] := ONHANDSTOCK;
I := I+1;
END;

```

```

Line 128 Col 1 Insert Indent A:OSFILE.PAS
RESET (OSFILE);
FOR I := 1 TO MAXITEM DO
BEGIN
READ (NAMEIN,NAMEREC);
CV[I] := NAMEREC.CODE;
END;
SCREEN_EDIT;
GOTOXY (1,10);
FOR I := 1 TO MAXITEM DO
BEGIN
RECNO := I;
WRITELN (RECNO:6,CV[I]:9);
GOTOXY (WHEREX+25,WHEREY-1);
WRITELN (OSV[RECNO]:12:3);
END;
GOTOXY (30,10);
REPEAT
WHILE NOT KEYSTAT (CH) DO BEGIN END;
IF NOT (CH IN COMMANDSET) THEN BEEP;
UNTIL (CH IN COMMANDSET);
CASE ORD(CH) OF
5 : BEGIN
FOR I := 1 TO MAXITEM DO
BEGIN

```

```

Line 148 Col 1 Insert Indent A:OSFILE.PAS
                READ (ONHANDSTOCK);
                RECNO := I;
                OSV [RECNO] := ONHANDSTOCK;
                GOTOXY (30,WHEREY+1);
                END;
            END;
        17 : I := MAXITEM;
        END;
SCREEN_MAIN;
RESET (OSFILE);
PI := 1;
WHILE NOT EOF(OSFILE) DO
BEGIN
    ONHANDSTOCK := OSV[PI];
    WRITE(OSFILE,ONHANDSTOCK);
    PI := PI+1;
END;
END;

```

```

'A' : BEGIN
    MAXREC := FILESIZE(NAMEIN)-FILESIZE(OSFILE);
    RECNO := FILESIZE(OSFILE);
    SEEK (OSFILE,RECNO);
    FOR I := 1 TO MAXREC DO

```

```

Line 171 Col 1 Insert Indent A:OSFILE.PAS
BEGIN
    OSV[I] := 0;
    ONHANDSTOCK :=0;
    WRITE (OSFILE,ONHANDSTOCK);
END;
SCREEN_MAIN;
GOTOXY (15,12);
WRITELN (' ') ,(MAXREC), ' Records Added');
END;

```

```

'p' : BEGIN
SCREEN_MAIN;GOTOXY(10,10);WRITELN(' >> Printing ...');
WRITELN(LST,'CODE':30,' ':10,'NAME', ' ON HAND STOCK ':42);
WRITELN(LST,'----':30,' ':10,'----','-----':42);
RESET(NAMEIN);
FOR I := 1 TO MAXITEM DO
BEGIN
    READ(OSFILE,ONHANDSTOCK);
    READ(NAMEIN,NAMEREC);
    WITH NAMEREC DO
        WRITELN(LST,CODE:30,' ':10,NAME:NAMELENGTH,' ':20-NAMELENGTH,
                ONHANDSTOCK:20:3,UNIT:6);WRITELN(LST);
END;
SCREEN_MAIN;

```

Line 194 Col 1 Insert Indent A:OSFILE.PAS
 END;

END;
 END;

BEGIN

```

SCREEN_MAIN;
ASSIGN (NAMEIN,'A:NAME.DAT');
ASSIGN (PERNAMEIN,'A:PERNAME.DAT');
ASSIGN (OSFILE,'A:OSFILE.DAT');
QUIT := FALSE;
REPEAT
  RESET (NAMEIN);
  RESET (PERNAMEIN);
  WHILE NOT KEYSTAT(CH) DO BEGIN END;
  IF CH IN COMMANDSET
    THEN CASE ORD(CH) OF
      1 : BEGIN
        RESET (OSFILE);
        BRANCH := 'A';
        PROC_EDIT(OSFILE,BRANCH,NAMEIN,NAMEREC,
          PERNAMEIN,PERNAMEREC);
        END;
      3 : BEGIN
        REWRITE (OSFILE);
  
```

Line 217 Col 1 Insert Indent A:OSFILE.PAS
 BRANCH := 'A';
 PROC_EDIT(OSFILE,BRANCH,NAMEIN,NAMEREC,
 PERNAMEIN,PERNAMEREC);

```

END;
5 : BEGIN
  RESET (OSFILE);
  BRANCH := 'E';
  PROC_EDIT(OSFILE,BRANCH,NAMEIN,NAMEREC,
    PERNAMEIN,PERNAMEREC);
END;
12 : BEGIN
  RESET (OSFILE);WRITELN;
  I := 1;
  WRITELN('NO':10,'ON HAND STOCK':20);WRITELN;
  WHILE NOT EOF(OSFILE) DO
  BEGIN
    READ(OSFILE,ONHANDSTOCK);
    WRITELN(I:10,ONHANDSTOCK:20:3);
    I := I+1;
  END;
  WRITELN;
  WRITE ('OSFILE':20,'Number of records : ':30,
    FILESIZE(OSFILE));
END;

```

```

Line 241 Col 1 Insert Indent A:OSFILE.PAS
      16 : BEGIN
            RESET (OSFILE);
            BRANCH := 'P';
            PROC_EDIT(OSFILE,BRANCH,NAMEIN,NAMEREC,
                    PERNAMEIN,PERNAMEREC);
            END;
      17 : QUIT := TRUE;
    END
  ELSE BEEP;
UNTIL QUIT;
CLOSE (NAMEIN);
CLOSE (PERNAMEIN);
CLOSE (OSFILE);
END.

```

Logged drive: A
Active directory: \

Work file: A:OSFILE.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 8394 bytes
Free: 54241 bytes

}

Compiling
254 lines

Code: 00B9 paragraphs (2960 bytes), 00C4 paragraphs free
Data: 000E paragraphs (224 bytes), 0FD9 paragraphs free
Stack/Heap: 314D paragraphs (201936 bytes)

```

Line 1 Col 1 Insert Indent A:CFILE.PAS
PROGRAM CAPACITYDATA;
TYPE

```

```

STRING4 = STRING[4];
STRING20 = STRING[20];
DEPTNAMERECORD= RECORD
    REC_NO : INTEGER;
    DEPTNAME: STRING20;
    NAMELENGTH : INTEGER;
END;
PERNAMERECORD = RECORD
    REC_NO : INTEGER;
    PERNAME : STRING20;
    NAMELENGTH : INTEGER;
END;
DEPTNAMEFILE = FILE OF DEPTNAMERECORD;
PERNAMEFILE = FILE OF PERNAMERECORD;
RECFILE = FILE OF REAL;

```

```
VAR
```

```

CH : CHAR;
I : INTEGER;
CAPACITY: REAL;
BRANCH : CHAR;
QUIT : BOOLEAN;
DEPTNAMEREC : DEPTNAMERECORD;

```

```
Line 33 Col 1 Insert Indent A:CFILE.PAS
```

```

DEPTNAMEIN : FILE OF DEPTNAMERECORD;
PERNAMEIN : FILE OF PERNAMERECORD;
CFILE : FILE OF REAL;

```

```
CONST
```

```
COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];
```

```

FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;
TYPE REG_PACK = RECORD
    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
END;
VAR REGISTERS : REG_PACK;
BEGIN
    REGISTERS.AX := $0600;
    REGISTERS.DX := 255;
    MSDOS(REGISTERS);
    CH :=CHR(REGISTERS.AX);
    KEYSTAT := CH (<) CHR(0)
END;

```



```

Line 70 Col 1 Insert Indent A:CFILE.PAS
PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;

```

```

PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');
  GOTOXY (12,2); WRITELN ('! CTRL-C : CLEAR !');
  GOTOXY (12,3); WRITELN ('! CTRL-A : APPEND !');
  GOTOXY (12,4); WRITELN ('! CTRL-E : EDIT !');
  GOTOXY (12,5); WRITELN ('! CTRL-P : PRINT !');
  GOTOXY (12,6); WRITELN ('! CTRL-Q : QUIT !');
END;

```

```

PROCEDURE SCREEN_EDIT;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');

Line 93 Col 1 Insert Indent A:CFILE.PAS
  GOTOXY (12,2); WRITELN ('! CTRL-E : EDIT !');
  GOTOXY (12,3); WRITELN ('! CTRL-S : SKIP PAGE !');
  GOTOXY (12,4); WRITELN ('! CTRL-Q : QUIT !');
  GOTOXY (5,8) ; WRITELN ('PERIOD : ');
  GOTOXY (3,12); WRITELN ('NO', ' ':6, 'PRODUCTION UNIT', ' ':4, 'CAPACITY');
END;

```

```

PROCEDURE PROC_EDIT (VAR CFILE : RECFILE;
  VAR BRANCH : CHAR;
  VAR DEPTNAMEIN : DEPTNAMEFILE;
  VAR DEPTNAMEREC : DEPTNAMERECORD;
  VAR PERNAMEIN : PERNAMEFILE;
  VAR PERNAMEREC : PERNAMERECORD);

```

```

TYPE
  INDEX = 1..100;
  STRING4 = STRING[4];
  STRING20 = STRING[20];
  DEPTVECTOR = ARRAY [INDEX] OF STRING20;
  PERNAMEVECTOR = ARRAY [INDEX] OF STRING20;
  CAPACITYVECTOR = ARRAY [INDEX] OF REAL;
  CAPACITYMATRIX = ARRAY [1..3,1..20] OF REAL;

```

Line 116 Col 1 Insert Indent A:CFILE.PAS

VAR

```

CH : CHAR;
NL : INTEGER;
DV : DEPTVECTOR;
PNV : PERNAMEVECTOR;
CPV : CAPACITYVECTOR;
C : CAPACITYMATRIX;
DI,PI,I : INDEX;
CAPACITY : REAL;
RECNO : INTEGER;
MAXREC : INTEGER;
MAXDEPT : INTEGER;
MAXPERIOD : INTEGER;
EXIT : BOOLEAN;

```

CONST

```

COMMANDSET : SET OF CHAR = [#17,#27,#19,#5];

```

BEGIN

```

MAXDEPT := FILESIZE(DEPTNAMEIN);
MAXPERIOD := FILESIZE(PERNAMEIN);
MAXREC := MAXDEPT*MAXPERIOD;

```

CASE BRANCH OF

'E' :BEGIN

I := 1;

WHILE NOT EOF (CFILE) DO

Line 139 Col 1 Insert Indent A:CFILE.PAS

BEGIN

READ (CFILE,CAPACITY);

CPV[I] := CAPACITY;

I := I+1;

END;

RESET (CFILE);

FOR I := 1 TO MAXDEPT DO

BEGIN

READ (DEPTNAMEIN,DEPTNAMEREC);

DV[I] := DEPTNAMEREC.DEPTNAME;

END;

PI := 1;

WHILE NOT EOF (PERNAMEIN) DO

BEGIN

SCREEN_EDIT;

READ (PERNAMEIN,PERNAMEREC);

PNV[PI] := PERNAMEREC.PERNAME;

GOTOXY (14,8);

WRITELN (PNV[PI]);

GOTOXY (1,14);

FOR I := 1 TO MAXDEPT DO

BEGIN

RECNO := (PI-1)*MAXDEPT + I;

WRITELN (RECNO:4,DV[I]:20);

Line 162 Col 1 Insert Indent A:CFILE.PAS

```

GOTOXY (WHEREX+25,WHEREY-1);
WRITELN (CPV[RECNO]:12:3);
END;
GOTOXY (30,14);
REPEAT
  WHILE NOT KEYSTAT (CH) DO BEGIN END;
  IF NOT (CH IN COMMANDSET) THEN BEEP;
UNTIL (CH IN COMMANDSET);
  CASE ORD(CH) OF
    5 : BEGIN
      FOR I := 1 TO MAXDEPT DO
        BEGIN
          READ (CAPACITY);
          RECNO := (PI-1)*MAXDEPT+I;
          CPV [RECNO] := CAPACITY;
          GOTOXY (30,WHEREY+1);
        END;
      END;
    17 : SEEK (PERNAMEIN,MAXPERIOD);
  END;
  PI := PI+1;
END;
SCREEN_MAIN;
RESET (CFILE);

```

Line 185 Col 1 Insert Indent A:CFILE.PAS

```

PI := 1;
WHILE NOT EOF(CFILE) DO
  BEGIN
    CAPACITY := CPV[PI];
    WRITE(CFILE,CAPACITY);
    PI := PI+1;
  END;
END;

```

'A' : BEGIN

```

MAXREC := FILESIZE(DEPTNAMEIN)*FILESIZE(PERNAMEIN)-FILESIZE(CFILE);
RECNO := FILESIZE(CFILE);
SEEK (CFILE,RECNO);
FOR I := 1 TO MAXREC DO
  BEGIN
    CPV[I] := 75;
    CAPACITY :=75;
    WRITE (CFILE,CAPACITY);
  END;
SCREEN_MAIN;
GOTOXY (15,12);
WRITELN ('> ',(MAXREC),' Records Added');
END;

```

Line 208 Col 1 Insert Indent A:CFILE.PAS

```
'P' : BEGIN
  SCREEN_MAIN;GOTOXY(10,10);WRITELN('> Printing ... ');
  WRITELN(LST,' ':45,'CAPACITY OF PRODUCTION UNITS');WRITELN(LST);
  WRITELN(LST);WRITE(LST,'PERIOD':10,' ':17);
  FOR PI := 1 TO MAXPERIOD DO
  BEGIN
    READ(PERNAMEIN,PERNAMEREC);
    WRITE(LST,PERNAMEREC.PERNAME:12);
    FOR DI := 1 TO MAXDEPT DO
      READ (CFILE,C[DI,PI]);
  END;
  WRITELN(LST);WRITE(LST,'-----':10,' ':17);
  FOR PI := 1 TO MAXPERIOD DO WRITE(LST,'-----':12);WRITELN(LST);
  FOR DI := 1 TO MAXDEPT DO
  BEGIN
    READ (DEPTNAMEIN,DEPTNAMEREC);
    NL := LENGTH(DEPTNAMEREC.DEPTNAME);
    WRITELN(LST);WRITELN(LST);WRITE(LST,' ':4,DEPTNAMEREC.DEPTNAME:NL,
      ' ':23-NL);
    FOR PI := 1 TO MAXPERIOD DO
      WRITE(LST,C[DI,PI]:12:3);
  END;
  WRITELN(LST);SCREEN_MAIN;
END;
```

Line 231 Col 1 Insert Indent A:CFILE.PAS

END;
END;

BEGIN

```
SCREEN_MAIN;
ASSIGN (DEPTNAMEIN,'A:DEPTNAME.DAT');
ASSIGN (PERNAMEIN,'A:PERNAME.DAT');
ASSIGN (CFILE,'A:CFILE.DAT');
QUIT := FALSE;
REPEAT
  RESET (DEPTNAMEIN);
  RESET (PERNAMEIN);
  WHILE NOT KEYSTAT(CH) DO BEGIN END;
  IF CH IN COMMANDSET
    THEN CASE ORD(CH) OF
      1 : BEGIN
        RESET (CFILE);
        BRANCH := 'A';
        PROC_EDIT(CFILE,BRANCH,DEPTNAMEIN,DEPTNAMEREC,
          PERNAMEIN,PERNAMEREC);
      END;
      3 : BEGIN
        REWRITE (CFILE);
        BRANCH := 'A';
```

```

Line 254 Col 1 Insert Indent A:CFILE.PAS
      PROC_EDIT(CFILE,BRANCH,DEPTNAMEIN,DEPTNAMEREC,
                PERNAMEIN,PERNAMEREC);
      END;
5 : BEGIN
      RESET (CFILE);
      BRANCH := 'E';
      PROC_EDIT(CFILE,BRANCH,DEPTNAMEIN,DEPTNAMEREC,
                PERNAMEIN,PERNAMEREC);
      END;
12 : BEGIN
      RESET (CFILE);
      I := 1;
      WHILE NOT EOF(CFILE) DO
      BEGIN
        READ(CFILE,CAPACITY);
        WRITELN(I,CAPACITY);
        I := I+1;
      END;
      END;
16 : BEGIN
      RESET (CFILE);
      BRANCH := 'P';
      PROC_EDIT(CFILE,BRANCH,DEPTNAMEIN,DEPTNAMEREC,
                PERNAMEIN,PERNAMEREC);
      BRANCH := 'P';
      PROC_EDIT(CFILE,BRANCH,DEPTNAMEIN,DEPTNAMEREC,
                PERNAMEIN,PERNAMEREC);

```

```

Line 289 Col 1 Insert Indent A:CFILE.PAS
      END;
      17 : QUIT := TRUE;
      END
      ELSE BEEP;
      UNTIL QUIT;
      CLOSE (DEPTNAMEIN);
      CLOSE (PERNAMEIN);
      CLOSE (CFILE);
      END.

```

Logged drive: A
Active directory: \

Work file: A:CFILE.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 8964 bytes
Free: 53671 bytes

)

Compiling
 275 lines

Code: 00D2 paragraphs (3360 bytes), 0CAB paragraphs free
Data: 000D paragraphs (208 bytes), 0FDA paragraphs free
Stack/Heap: 3111 paragraphs (200976 bytes)

)

```

Line 1 Col 1 Insert Indent A:OFILE.PAS
PROGRAM OPTIMEDATA;
TYPE

```

```

STRING4 = STRING[4];
STRING20 = STRING[20];
NAMERECORD = RECORD
    REC_NO : INTEGER;
    CODE : STRING4;
    NAME : STRING20;
    UNIT : STRING4;
    CLASS : STRING4;
    NAMELENGTH : INTEGER;
END;
DEPTNAMERECORD = RECORD
    REC_NO : INTEGER;
    DEPTNAME : STRING20;
    NAMELENGTH : INTEGER;
END;
NAMEFILE = FILE OF NAMERECORD;
DEPTNAMEFILE = FILE OF DEPTNAMERECORD;
RECFILE = FILE OF REAL;

```

```

VAR

```

```

CH : CHAR;
I : INTEGER;
OPTIME : REAL;

```

```

Line 35 Col 1 Insert Indent A:OFILE.PAS

```

```

BRANCH : CHAR;
QUIT : BOOLEAN;
NAMEREC : NAMERECORD;
DEPTNAMEREC : DEPTNAMERECORD;
NAMEIN : FILE OF NAMERECORD;
DEPTNAMEIN : FILE OF DEPTNAMERECORD;
OFILE : FILE OF REAL;

```

```

CONST

```

```

COMMANDSET : SET OF CHAR = [#1,#3,#5,#12,#16,#17];

```

```

FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;

```

```

TYPE REG_PACK = RECORD

```

```

    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
END;

```

```

VAR REGISTERS : REG_PACK;

```

```

BEGIN

```

```

    REGISTERS.AX := $0600;
    REGISTERS.DX := 255;
    MSDOS(REGISTERS);
    CH :=CHR(REGISTERS.AX);
    KEYSTAT := CH () CHR(0)

```

END;

Line 50 Col 1 Insert Indent A:OFILE.PAS

```
PROCEDURE BEEP;
BEGIN
  WRITE(CHR(7));
END;
```

```
PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('!   ***** MENU *****   !');
  GOTOXY (12,2); WRITELN ('!   CTRL-C : CLEAR           !');
  GOTOXY (12,3); WRITELN ('!   CTRL-A : APPEND         !');
  GOTOXY (12,4); WRITELN ('!   CTRL-E : EDIT           !');
  GOTOXY (12,5); WRITELN ('!   CTRL-P : PRINT          !');
  GOTOXY (12,6); WRITELN ('!   CTRL-Q : QUIT           !');
END;
```

Line 73 Col 1 Insert Indent A:OFILE.PAS

```
PROCEDURE SCREEN_EDIT;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('!   ***** MENU *****   !');
  GOTOXY (12,2); WRITELN ('!   CTRL-E : EDIT           !');
  GOTOXY (12,3); WRITELN ('!   CTRL-S : SKIP PAGE     !');
  GOTOXY (12,4); WRITELN ('!   CTRL-Q : QUIT           !');
  GOTOXY (5,7) ; WRITELN ('PRODUCTION UNIT : ');
  GOTOXY (5,9); WRITELN ('NO', ' ':5, 'CODE', ' ':12, 'OPTIME');
END;
```

```
PROCEDURE PROC_EDIT (VAR OFILE      : RECFILE;
                     VAR BRANCH     : CHAR;
                     VAR NAMEIN     : NAMEFILE;
                     VAR NAMEREC    : NAMERECORD;
                     VAR DEPTNAMEIN : DEPTNAMEFILE;
                     VAR DEPTNAMEREC: DEPTNAMERECORD);
```

```
TYPE
  INDEX   = 1..100;
  STRING4 = STRING[4];
  STRING20 = STRING[20];
```



```

Line 97 Col 1 Insert Indent A:OFILE.PAS
CODEVECTOR = ARRAY [INDEX] OF STRING4;
DEPTNAMEVECTOR = ARRAY [INDEX] OF STRING20;
OPTIMEVECTOR = ARRAY [INDEX] OF REAL;
OPTINEMATRIX = ARRAY [1..3,1..20] OF REAL;
VAR
CH : CHAR;
I,DI: INDEX;
CV : CODEVECTOR;
DNV : DEPTNAMEVECTOR;
OV : OPTIMEVECTOR;
O : OPTINEMATRIX;
NL : INTEGER;
OPTIME : REAL;
RECNO : INTEGER;
MAXREC : INTEGER;
MAXITEM : INTEGER;
MAXDEPT : INTEGER;
EXIT : BOOLEAN;
CONST
COMMANDSET : SET OF CHAR = [#17,#27,#19,#5];
BEGIN
MAXITEM := FILESIZE(NAMEIN);
MAXDEPT := FILESIZE(DEPTNAMEIN);
MAXREC := MAXITEM*MAXDEPT;

```

```

Line 121 Col 1 Insert Indent A:OFILE.PAS
CASE BRANCH OF
'E' :BEGIN
I := 1;
WHILE NOT EOF (OFILE) DO
BEGIN
READ (OFILE,OPTIME);
OV[I] := OPTIME;
I := I+1;
END;
RESET (OFILE);
FOR I := 1 TO MAXITEM DO
BEGIN
READ (NAMEIN,NAMEREC);
CV[I] := NAMEREC.CODE;
END;
DI := 1;
WHILE NOT EOF (DEPTNAMEIN) DO
BEGIN
SCREEN_EDIT;
READ (DEPTNAMEIN,DEPTNAMEREC);
DNV[DI] := DEPTNAMEREC.DEPTNAME;
GOTOXY (25,7);
WRITELN (DNV[DI]);
GOTOXY (1,11);

```

Line 145 Col 1 Insert Indent A:OFILE.PAS

```

FOR I := 1 TO MAXITEM DO
  BEGIN
    RECNO := (DI-1)*MAXITEM + I;
    WRITELN (RECNO:6,CV[I]:9);
    GOTOXY (WHEREX+20,WHEREY-1);
    WRITELN (OV[RECNO]:12:3);
  END;
  GOTOXY (25,11);
  REPEAT
    WHILE NOT KEYSTAT (CH) DO BEGIN END;
    IF NOT (CH IN COMMANDSET) THEN BEEP;
  UNTIL (CH IN COMMANDSET);
  CASE ORD(CH) OF
    5 : BEGIN
      FOR I := 1 TO MAXITEM DO
        BEGIN
          READ (OPTIME);
          RECNO := (DI-1)*MAXITEM+I;
          OV [RECNO] := OPTIME;
          GOTOXY (25,WHEREY+1);
        END;
      END;
    17 : SEEK (DEPTNAMEIN,MAXDEPT);
  END;

```

Line 169 Col 1 Insert Indent A:OFILE.PAS

```

DI := DI+1;
END;
SCREEN_MAIN;
RESET (OFILE);
DI := 1;
WHILE NOT EOF(OFILE) DO
  BEGIN
    OPTIME := OV[DI];
    WRITE(OFILE,OPTIME);
    DI := DI+1;
  END;
END;

```

'A' : BEGIN

```

MAXREC := FILESIZE(NAMEIN)*FILESIZE(DEPTNAMEIN)-FILESIZE(OFILE);
RECNO := FILESIZE(OFILE);
SEEK (OFILE,RECNO);
FOR I := 1 TO MAXREC DO
  BEGIN
    OV[I] := 0;
    OPTIME :=0;
    WRITE (OFILE,OPTIME);
  END;
SCREEN_MAIN;

```

Line 193 Col 1 Insert Indent A:OFILE.PAS

```
GOTOXY (15,12);
WRITELN (') ',(MAXREC),' Records Added');
END;
```

'P': BEGIN

```
SCREEN_MAIN;GOTOXY(10,10);WRITELN(') Printing ... ');
WRITELN(LST,' ):37,'OPERATION TIMES AT PRODUCTION UNITS');
WRITELN(LST);WRITELN(LST,' ):70,'(HOURS)');
WRITELN(LST);WRITE(LST,'NAME':10,' ):25);
FOR DI := 1 TO MAXDEPT DO
BEGIN
  READ(DEPTNAMEIN,DEPTNAMEREC);
  WRITE(LST,DEPTNAMEREC.DEPTNAME:20);
  FOR I := 1 TO MAXITEM DO
    READ(OFILE,O[DI,I]);
  END;
  WRITELN(LST);WRITE(LST,'----':10,' ):25);
  FOR DI := 1 TO MAXDEPT DO WRITE(LST,'-----':20);
  WRITELN(LST);
  FOR I := 1 TO MAXITEM DO
  BEGIN
    READ (NAMEIN,NAMEREC);
    NL := LENGTH(NAMEREC.NAME);
    WRITELN(LST);WRITELN(LST);
```

Line 217 Col 1 Insert Indent A:OFILE.PAS

```
WRITE(LST,' ):6,NAMEREC.NAME:NL,' ):21-NL);
FOR DI := 1 TO MAXDEPT DO
  WRITE(LST,O[DI,I]:20:3);
END;
WRITELN(LST);SCREEN_MAIN;
END;
```

END;
END;

BEGIN

```
SCREEN_MAIN;
ASSIGN (NAMEIN,'A:NAME.DAT');
ASSIGN (DEPTNAMEIN,'A:DEPTNAME.DAT');
ASSIGN (OFILE,'A:OFILE.DAT');
QUIT := FALSE;
REPEAT
  RESET (NAMEIN);
  RESET (DEPTNAMEIN);
  WHILE NOT KEYSTAT(CH) DO BEGIN END;
  IF CH IN COMMANDSET
    THEN CASE ORD(CH) OF
      1 : BEGIN
          RESET (OFILE);
          BRANCH := 'A';
```

```

Line 241 Col 1 Insert Indent A:OFILE.PAS
      PROC_EDIT(OFILE,BRANCH,NAMEIN,NAMEREC,
                DEPTNAMEIN,DEPTNAMEREC);
      END;
3 : BEGIN
      REWRITE (OFILE);
      BRANCH := 'A';
      PROC_EDIT(OFILE,BRANCH,NAMEIN,NAMEREC,
                DEPTNAMEIN,DEPTNAMEREC);
      END;
5 : BEGIN
      RESET (OFILE);
      BRANCH := 'E';
      PROC_EDIT(OFILE,BRANCH,NAMEIN,NAMEREC,
                DEPTNAMEIN,DEPTNAMEREC);
      END;
12 : BEGIN
      RESET (OFILE);
      I := 1;
      WHILE NOT EOF(OFILE) DO
      BEGIN
        READ(OFILE,OPTIME);
        WRITELN(I,OPTIME);
        I := I+1;
      END;

```

```

Line 287 Col 1 Insert Indent A:OFILE.PAS
      END;
16 : BEGIN
      RESET (OFILE);
      BRANCH := 'P';
      PROC_EDIT(OFILE,BRANCH,NAMEIN,NAMEREC,
                DEPTNAMEIN,DEPTNAMEREC);
      END;
17 : QUIT := TRUE;
      END
      ELSE BEEP;
      UNTIL QUIT;
CLOSE (NAMEIN);
CLOSE (DEPTNAMEIN);
CLOSE (OFILE);
END.

```

Logged drive: A
Active directory: \

Work file: A:OFILE.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 8971 bytes
Free: 53664 bytes

)

Compiling
 279 lines

Code: 00D1 paragraphs (3344 bytes), 0C4C paragraphs free
Data: 000E paragraphs (224 bytes), 0FD9 paragraphs free
Stack/Heap: 3111 paragraphs (200976 bytes)

)

```

Line 1  Col 1  Insert  Indent  A:ABCANALY.PAS
PROGRAM ABCANALYSIS;
CONST
  CODESIZE = 4;
  ALIMIT = 0.7;
  BLIMIT = 0.95;
TYPE
  INDEX = 1..20;
  STRING4 = STRING[4];
  STRING20 = STRING[20];
  NAMERECORD = RECORD
    REC_NO : INTEGER;
    CODE : STRING4;
    NAME : STRING20;
    UNIT : STRING4;
    CLASS : STRING4;
    NAMELENGTH : INTEGER;
  END;
  ABCINRECORD = RECORD
    REC_NO : INTEGER;
    CODE : STRING4;
    PRICE : REAL;
    USAGE : REAL;
  END;
  ABCOUTRECORD = RECORD

```

```

Line 25  Col 1  Insert  Indent  A:ABCANALY.PAS
  CODE : STRING4;
  PRICE : REAL;
  USAGE : REAL;
  ABCCODE : CHAR;
  END;
NAMEVECTOR = ARRAY [INDEX] OF STRING20;
CODEVECTOR = ARRAY [INDEX] OF STRING4;
UNITVECTOR = ARRAY [INDEX] OF STRING4;
CLASSVECTOR = ARRAY [INDEX] OF STRING4;
PRICEVECTOR = ARRAY [INDEX] OF REAL;
USAGEVECTOR = ARRAY [INDEX] OF REAL;
PERCENTVECTOR = ARRAY [INDEX] OF REAL;
WORKVECTOR = ARRAY [INDEX] OF REAL;
ABCCODEVECTOR = ARRAY [INDEX] OF CHAR;
VAR
  RECNO, MAXREC : INTEGER;
  I, J, JJ, K, NL : INTEGER;
  SUM, CUM_SUM : REAL;
  QUIT : BOOLEAN;
  CH : CHAR;
  COD: STRING4;
  TC : STRING4;
  TI : REAL;
  CV : CODEVECTOR;

```

```

Line 49 Col 1 Insert Indent A:ABCANALY.PAS
MV : NAMEVECTOR;
BV : UNITVECTOR;
KV : CLASSVECTOR;
PV : PRICEVECTOR;
UV : USAGEVECTOR;
WV : WORKVECTOR;
YV : PERCENTVECTOR;
ABC : ABCCODEVECTOR;
NAMEREC: NAMERECORD;
INREC : ABCINRECORD;
OUTREC : ABCOUTRECORD;
NAMEIN : FILE OF NAMERECORD;
ABCIN : FILE OF ABCINRECORD;
ABCOU : FILE OF ABCOUTRECORD;
CONST COMMANDSET : SET OF CHAR = [#12,#16,#17,#18];

```

```

FUNCTION KEYSTAT (VAR CH : CHAR) : BOOLEAN;
TYPE REG_PACK = RECORD
    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
    END;
VAR REGISTERS : REG_PACK;
BEGIN

```

```

Line 73 Col 1 Insert Indent A:ABCANALY.PAS
REGISTERS.AX := $0600;
REGISTERS.DX := 255;
MSDOS (REGISTERS);
CH := CHR (REGISTERS.AX);
KEYSTAT := CH <> CHR(0)
END;

```

```

PROCEDURE SCREEN_MAIN;
BEGIN
    CLRSCR;
    GOTOXY (12,1); WRITELN ('| **** MENU **** |');
    GOTOXY (12,2); WRITELN ('| CTRL-R : RUN |');
    GOTOXY (12,3); WRITELN ('| CTRL-L : LIST |');
    GOTOXY (12,4); WRITELN ('| CTRL-P : PRINT |');
    GOTOXY (12,5); WRITELN ('| CTRL-Q : QUIT |');
END;

```

```

BEGIN
    QUIT := FALSE;

```

```

Line 97 Col 1 Insert Indent A:ABCANALY.PAS
SCREEN_MAIN;
ASSIGN (NAMEIN,'A:NAME.DAT');
ASSIGN (ABCIN,'A:ABCIN.DAT');
ASSIGN (ABCOUT,'A:ABCOUT.DAT');
RESET (NAMEIN);
RESET (ABCIN);
MAXREC := FILESIZE(NAMEIN);
I := 1;
SUM := 0;
CUM_SUM := 0;
REPEAT
  WHILE NOT KEYSTAT (CH) DO BEGIN END;
  IF NOT (CH IN COMMANDSET) THEN WRITELN (CHR(7));
UNTIL (CH IN COMMANDSET);
CASE ORD(CH) OF
17 : QUIT := TRUE;
18 : BEGIN
  CLRSCR;
  WRITELN(' :20,'ABC ANALYSIS FOR INVENTORY ITEMS');
  WRITELN;WRITELN;
  WRITELN('NO', ' :2,'CODE', ' :6,'NAME', ' :11,'UNIT',
          ' :3,'CLASS', ' :2,'PURCHASE PRICE', ' :2,
          'ANNUAL USAGE', ' :2,'ABCCODE');
  REWRITE (ABCOUT);

```

```

Line 121 Col 1 Insert Indent A:ABCANALY.PAS
WHILE NOT EOF(ABCIN) DO
BEGIN
  READ(ABCIN,INREC);
  WITH INREC DO
  BEGIN
    CV[I] := CODE;
    PV[I] := PRICE;
    UV[I] := USAGE;
  END;
  WV[I] := PV[I] * UV[I];
  SUM := SUM + WV[I];
  I := I+1;
END;
FOR I := 1 TO MAXREC DO YV[I] := WV[I] /SUM;
FOR I := 1 TO MAXREC DO
BEGIN
  J := I;
  JJ := I+1;
  FOR K := JJ TO MAXREC DO
  IF WV[K] > WV[J]
  THEN J := K;
  TC := CV[I];
  CV[I] := CV[J];
  CV[J] := TC;

```


Line 145 Col 1 Insert Indent A:ABCANALY.PAS

```

    TI := PV[I];
    PV[I] := PV[J];
    PV[J] := TI;
    TI := UV[I];
    UV[I] := UV[J];
    UV[J] := TI;
    TI := WV[I];
    WV[I] := WV[J];
    WV[J] := TI;
END;
FOR I := 1 TO MAXREC DO
BEGIN
    CUM_SUM := CUM_SUM + YV[I];
    IF CUM_SUM (<= ALIMIT
    THEN ABC[I] := 'A'
    ELSE IF CUM_SUM (<= BLIMIT
    THEN ABC[I] := 'B'
    ELSE ABC[I] := 'C';
END;
I := 1;
J := 1;
WHILE NOT EOF (NAMEIN) DO
BEGIN
    READ(NAMEIN,NAMEREC);

```

Line 169 Col 1 Insert Indent A:ABCANALY.PAS

```

WITH NAMEREC DO
BEGIN
    RECNO := REC_NO;
    COD := CODE;
    NV[J] := NAME;
    BV[J] := UNIT;
    KV[J] := CLASS;
END;
FOR I := 1 TO MAXREC DO
BEGIN
    IF COD = CV[I]
    THEN BEGIN
        WITH OUTREC DO
        BEGIN
            CODE := CV[I];
            PRICE := PV[I];
            USAGE := UV[I];
            ABCCODE := ABC[I];
            WRITE (ABCCOUT,OUTREC);
            NL := NAMEREC.NAMELENGTH + 2;
            WRITELN;WRITELN (RECNO:2,CV[I]:6,NV[J]:NL,
                ' ':21-NL,BV[J]:4,KV[J]:6,PV[I]:16:2,
                UV[I]:14:2,ABCCODE:8);
            WRITELN(LST);WRITELN(LST,RECNO:2,CV[I]:6,

```

```

Line 193 Col 1 Insert Indent A:ABCANALY.PAS
                        NV[J]:NL, ' ':21-NL,BV[J]:4,KV[J]:6,PV[I]:16:
                        UV[I]:14:2,ABCCODE:8);
                                END;
                                END;
                                END;
                                J := J+1;
                                END;
                                END;
12 : BEGIN
    RESET (ABCOU);
    CLRSCR;
    WRITELN;WRITELN(' ':5,'NO', ' ':5,'CODE', ' ':5,'PURCHASE PRICE',
                    ' ':5,'ANNUAL USAGE', ' ':5,'ABCCODE');WRITELN;
    FOR I := 1 TO MAXREC DO
    BEGIN
        READ (ABCOU,OUTREC);
        WITH OUTREC DO
            WRITELN(I:7,CODE:9,PRICE:17:3,USAGE:18:3,ABCCODE:10);
        END;
        WRITELN;WRITE('ABCOU File ':20,'Number of Records : ':30,
        FILESIZE(ABCOU));
    END;
16 : BEGIN
    SCREEN_MAIN;GOTOXY(10,10);WRITELN(') Printing ...');

```

```

Line 217 Col 1 Insert Indent A:ABCANALY.PAS
    RESET (ABCOU);
    RESET (NAMEIN);
    WRITELN(LST,' ':50,'ABC ANALYSIS FOR INVENTORY ITEMS');
    WRITELN(LST);WRITELN(LST);
    WRITELN(LST,'NO':15,'CODE':10, ' ':7,'NAME', 'UNIT':29,'CLASS':15,
            'PURCHASE PRICE':20,'ANNUAL USAGE':20,'ABCCODE':10);
    WRITELN(LST,'--':15,'----':10, ' ':7,'----', '----':29,'----':15,
            '-----':20,'-----':20,'-----':10);
    FOR I := 1 TO MAXREC DO
    BEGIN
        READ (ABCOU,OUTREC);
        READ(NAMEIN,NAMEREC);
        WITH OUTREC DO
        BEGIN
            PV[I] := PRICE;
            UV[I] := USAGE;
            ABC[I]:= ABCCODE;
        END;
        WITH NAMEREC DO
        BEGIN
            RECNO := REC_NO;
            CV[I] := CODE;
            NV[I] := NAME;
            BV[I] := UNIT;

```

```
Line 263 Col 1 Insert Indent A:ABCANALY.PAS
  KV[I] := CLASS;
  NL    := NAMELENGTH+2;
  END;
  WRITELN(LST);
  WRITELN(LST,RECNO:15,CV[I]:10,' ':5,NV[I]:NL,' ':30-NL,BV[I]:4,
    KV[I]:14,PV[I]:20:2,UV[I]:21:2,ABC[I]:8);
  END;
  SCREEN_MAIN;
  END;
  END;
  CLOSE (NAMEIN);
  CLOSE (ABCIN);
  CLOSE (ABCOU);
END.
```

Logged drive: A
Active directory: \

Work file: A:ABCANALY.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 8052 bytes
Free: 54583 bytes

)

Compiling
254 lines

Code: 010A paragraphs (4256 bytes), 0C73 paragraphs free
Data: 005D paragraphs (1488 bytes), 0F8A paragraphs free
Stack/Heap: 30C2 paragraphs (199712 bytes)

)

```

Line 1 Col 1 Insert Indent A:INV-PLAN.PAS
PROGRAM INVENTORY_PLANNING;
CONST
  INTEREST = 0.65;
  NA = 0.5;
  NB = 1.0;
  NC = 1.5;

```

```
TYPE
```

```

INDEX = 1..20;
STRING4 = STRING[4];
STRING20 = STRING[20];
NAMERECORD = RECORD
  REC_NO : INTEGER;
  CODE   : STRING4;
  NAME   : STRING20;
  UNIT   : STRING4;
  CLASS  : STRING4;
  NAMELENGTH : INTEGER;
END;
ABCRECORD = RECORD
  CODE   : STRING4;
  PRICE  : REAL;
  USAGE  : REAL;
  ABCCODE : CHAR;
END;

```

```

Line 25 Col 1 Insert Indent A:INV-PLAN.PAS
PURCHRECORD = RECORD
  REC_NO      : INTEGER;
  CODE        : STRING4;
  REPLCOST    : REAL;
  LEADTIME    : INTEGER;
  LEADVARIANCE : REAL;
END;
ORDERRECORD = RECORD
  CODE        : STRING4;
  ORDQUANT    : REAL;
  LEADTIME    : INTEGER;
  SSTOCK      : REAL;
END;

NAMEVECTOR   = ARRAY [INDEX] OF STRING20;
CODEVECTOR   = ARRAY [INDEX] OF STRING4;
UNITVECTOR   = ARRAY [INDEX] OF STRING4;
CLASSVECTOR  = ARRAY [INDEX] OF STRING4;
NAMELENGTHVECTOR = ARRAY [INDEX] OF INTEGER;
PRICEVECTOR  = ARRAY [INDEX] OF REAL;
USAGEVECTOR  = ARRAY [INDEX] OF REAL;
ABCVECTOR    = ARRAY [INDEX] OF CHAR;
REPLCOSTVECTOR = ARRAY [INDEX] OF REAL;
LEADTIMEVECTOR = ARRAY [INDEX] OF INTEGER;
LEADVARIANCEVECTOR = ARRAY [INDEX] OF REAL;

```

Line 49 Col 1 Insert Indent A:INV-PLAN.PAS

```
ORDERQUANTITYVECTOR = ARRAY [INDEX] OF REAL;
SAFETYSTOCKVECTOR   = ARRAY [INDEX] OF REAL;
```

VAR

```
I, MAXITEM : INTEGER;
QUIT: BOOLEAN;
N  : REAL;
CH  : CHAR;
NR  : STRING4;
CV  : CODEVECTOR;
NV  : NAMEVECTOR;
BV  : UNITVECTOR;
KV  : CLASSVECTOR;
NL  : NAMELENGTHVECTOR;
PV  : PRICEVECTOR;
UV  : USAGEVECTOR;
ABC : ABCVECTOR;
RCV : REPLCOSTVECTOR;
LTV : LEADTIMEVECTOR;
LVV : LEADVARIANCEVECTOR;
QV  : ORDERQUANTITYVECTOR;
SV  : SAFETYSTOCKVECTOR;
NAMEREC : NAMERECORD;
ABCREC  : ABCRECORD;
PURCHREC : PURCHRECORD;
```

Line 73 Col 1 Insert Indent A:INV-PLAN.PAS

```
OUTREC : ORDERRECORD;
NAMEIN  : FILE OF NAMERECORD;
ABCOUT  : FILE OF ABCRECORD;
INVIN   : FILE OF PURCHRECORD;
INVOUT  : FILE OF ORDERRECORD;
CONST COMMANDSET : SET OF CHAR = [#5,#16,#17,#18];
```

```
FUNCTION KEYSTAT (VAR CH : CHAR) : BOOLEAN;
TYPE REG_PACK = RECORD
    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
END;
VAR REGISTERS : REG_PACK;
BEGIN
    REGISTERS.AX := $0600;
    REGISTERS.DX := 255;
    MSDOS (REGISTERS);
    CH := CHR (REGISTERS.AX);
    KEYSTAT := CH (<) CHR(0)
END;
```

```

Line 106 Col 1 Insert Indent A:INV-PLAN.PAS
PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('! **** MENU **** !');
  GOTOXY (12,2); WRITELN ('! CTRL-R : RUN !');
  GOTOXY (12,3); WRITELN ('! CTRL-E : EDIT !');
  GOTOXY (12,4); WRITELN ('! CTRL-P : PRINT !');
  GOTOXY (12,5); WRITELN ('! CTRL-Q : QUIT !');
END;

```

```

BEGIN
  QUIT := FALSE;
  SCREEN_MAIN;
  ASSIGN (NAMEIN, 'A:NAME.DAT');
  ASSIGN (ABCOU, 'A:ABCOU.DAT');
  ASSIGN (INVIN, 'A:INVIN.DAT');
  ASSIGN (INVOUT, 'A:INVOUT.DAT');
  RESET (NAMEIN);
  RESET (ABCOU);
  RESET (INVIN);
  MAXITEM := FILESIZE (NAMEIN);
  FOR I := 1 TO MAXITEM DO

```

```

Line 129 Col 1 Insert Indent A:INV-PLAN.PAS
BEGIN
  READ (NAMEIN, NAMEREC);
  WITH NAMEREC DO
  BEGIN
    CV[I] := CODE;
    NV[I] := NAME;
    BV[I] := UNIT;
    KV[I] := CLASS;
    NL[I] := NAMELENGTH + 2;
  END;
END;
FOR I := 1 TO MAXITEM DO
BEGIN
  READ (ABCOU, ABCREC);
  WITH ABCREC DO
  BEGIN
    PV[I] := PRICE;
    UV[I] := USAGE;
    ABC[I] := ABCCODE;
  END;
END;
FOR I := 1 TO MAXITEM DO
BEGIN
  READ (INVIN, PURCHREC);

```

```

Line 152 Col 1 Insert Indent A:INV-PLAN.PAS
WITH PURCHREC DO
BEGIN
  RCV[I] := REPLCOST;
  LTV[I] := LEADTIME;
  LVV[I] := LEADVARIANCE;
END;
END;
REPEAT
  WHILE NOT KEYSTAT (CH) DO BEGIN END;
  IF NOT (CH IN COMMANDSET) THEN WRITELN(CHR(7));
UNTIL (CH IN COMMANDSET);
IF NOT (ORD(CH) =17) THEN
BEGIN
CASE ORD(CH) OF
18 : BEGIN
  CLRSCR;
  WRITELN (' ':25,'INVENTORY PLANNING');
  WRITELN;WRITELN;
  WRITELN ('NO':2,'CODE':6,'NAME':10,'UNIT':15,'CLASS':7,
           'ORDER QUANTITY':16,'LEAD TIME':11,'SAFETY STOCK':13);
  REWRITE (INVOUT);
  FOR I := 1 TO MAXITEM DO
  BEGIN
    QV[I] := SQRT(2*RCV[I]*UV[I]/INTEREST/PV[I]);

```

```

Line 175 Col 1 Insert Indent A:INV-PLAN.PAS
  IF ABC[I] = 'A' THEN N := NA;
  IF ABC[I] = 'B' THEN N := NB;
  IF ABC[I] = 'C' THEN N := NC;
  IF ((NOT(N = NA)) AND (NOT (N = NB)) AND (NOT (N = NC)))
  THEN WRITELN ('*** ABC CODE ERROR ***');
  SV[I] := N*SQRT(LVV[I])*UV[I]/12/3;
  WITH OUTREC DO
  BEGIN
    CODE      := CV[I];
    ORDQUANT := QV[I];
    LEADTIME := LTV[I];
    SSTOCK   := SV[I];
  END;
  WRITE (INVOUT,OUTREC);
  WRITELN (1:2,CV[I]:6,NV[I]:NL[I],' ':21-NL[I],BV[I]:4,
           KV[I]:5,QV[I]:15:3,LTV[I]:10,SV[I]:16:3);
  END;
END;
5 : BEGIN
  RESET (INVOUT);
  CLRSCR;
  GOTOXY (12,1); WRITELN ('' **** MENU **** '');
  GOTOXY (12,2); WRITELN ('' CTRL-E : EDIT '');
  GOTOXY (5,7);

```

```

Line 198 Col 1 Insert Indent A:INV-PLAN.PAS
  WRITELN ('NO':6,'CODE':8,'ORDER QUANTITY':18,
    'LEAD TIME':13,'SAFETY STOCK':16);WRITELN;
  FOR I := 1 TO MAXITEM DO
  BEGIN
    READ (INVOUT,OUTREC);
    WITH OUTREC DO
    WRITELN(I:10,CODE:8,ORDQUANT:15:3,LEADTIME:12,SSTOCK:19:3);
    WHILE NOT KEYSTAT(CH) DO BEGIN END;
    IF (ORD(CH)=5)
    THEN BEGIN
      GOTOXY(WHEREX+25,WHEREY-1);READ(OUTREC.ORDQUANT);
      GOTOXY(WHEREX+28,WHEREY);READ(OUTREC.SSTOCK);
      WRITELN;
      SEEK (INVOUT,I-1);
      WRITE(INVOUT,OUTREC);
    END;
  END;
  WRITELN;WRITE('INVOUT File ':20,'Number of Records : ':30,
    FILESIZE(INVOUT));
END;
16 : BEGIN
  SCREEN_MAIN;GOTOXY(10,10);WRITELN('>> Printing ...');
  WRITELN (LST,' ':60,'INVENTORY PLANNING');
  WRITELN(LST);WRITELN(LST);

Line 221 Col 1 Insert Indent A:INV-PLAN.PAS
  WRITELN(LST,'NO':15,'CODE':10,' ':7,'NAME','UNIT':29,'CLASS':10,
    'ORDER QUANTITY':19,'LEAD TIME':14,'SAFETY STOCK':16);
  WRITELN(LST,'--':15,'----':10,' ':7,'-----','-----':29,'-----':10,
    '-----':19,'-----':14,'-----':16);
  RESET(INVOUT);
  FOR I := 1 TO MAXITEM DO
  BEGIN
    READ(INVOUT,OUTREC);
    WRITELN(LST);
    WRITELN(LST,I:15,CV[I]:10,' ':5,NV[I]:NL[I],' ':30-NL[I],BV[I]:4,
      KV[I]:9,OUTREC.ORDQUANT:18:3,OUTREC.LEADTIME:13,OUTREC.SSTOCK:18:3);
  END;
  SCREEN_MAIN;
END;
17 : QUIT := TRUE;
END;
END;
CLOSE (NAMEIN);
CLOSE (ABCCUT);
CLOSE (INVIN);
CLOSE (INVOUT);
END.

```


Logged drive: A
Active directory: \

Work file: A:INV-PLAN.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 7607 bytes
Free: 55028 bytes

)

Compiling
 238 lines

Code: 00E1 paragraphs (3600 bytes), 0C9C paragraphs free
Data: 0074 paragraphs (1856 bytes), 0F73 paragraphs free
Stack/Heap: 30F0 paragraphs (200448 bytes)

)

Line 1 Col 1 Insert Indent A:GOESINTO.PAS

PROGRAM GOESINTO;

TYPE

```

INDEX = 1..20;
STRING4 = STRING[4];
STRING20 = STRING[20];
NAMERECORD = RECORD
    REC_NO : INTEGER;
    CODE   : STRING4;
    NAME   : STRING20;
    UNIT   : STRING4;
    CLASS  : STRING4;
    NAMELENGTH : INTEGER;
END;
BOMRECORD = RECORD
    REC_NO      : INTEGER;
    FATHERCODE  : STRING4;
    SONCODE     : STRING4;
    GOESINTO    : REAL;
END;
RECNOVECTOR   = ARRAY [INDEX] OF INTEGER;
CODEVECTOR    = ARRAY [INDEX] OF STRING4;
FATHERCODEVECTOR = ARRAY [INDEX] OF STRING4;
SONCODEVECTOR = ARRAY [INDEX] OF STRING4;
GOESINTOVECTOR = ARRAY [INDEX] OF REAL;

```

Line 25 Col 1 Insert Indent A:GOESINTO.PAS

```

WORKMATRIX   = ARRAY [INDEX,INDEX] OF REAL;
MATERIALMATRIX = ARRAY [INDEX,INDEX] OF REAL;
WORKVECTOR   = ARRAY [INDEX] OF REAL;

```

VAR

```

I,J,RI,CI,K,JMAX : INDEX;
IMAX : INTEGER;
C : REAL;
CH : CHAR;
RV : RECNOVECTOR;
CV : CODEVECTOR;
FCV : FATHERCODEVECTOR;
SCV : SONCODEVECTOR;
GIV : GOESINTOVECTOR;
WM : WORKMATRIX;
MM : MATERIALMATRIX;
WV : WORKVECTOR;
QUIT : BOOLEAN;
BOMREC : BOMRECORD;
NAMEREC : NAMERECORD;
NAMEIN : FILE OF NAMERECORD;
BOMFILE : FILE OF BOMRECORD;
TFILE : FILE OF REAL;
UFILE : FILE OF REAL;

```

CONST

Line 49 Col 1 Insert Indent A:GOESINTO.PAS
 COMADSET : SET OF CHAR = [#12,#16,#17,#18];

```
FUNCTION KEYSTAT(VAR CH : CHAR) : BOOLEAN;
TYPE REG_PACK = RECORD
    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
END;
VAR REGISTERS : REG_PACK;
BEGIN
    REGISTERS.AX := $0600;
    REGISTERS.DX := 255;
    MSDOS(REGISTERS);
    CH := CHR(REGISTERS.AX);
    KEYSTAT := CH <>CHR(0)
END;
```

```
PROCEDURE BEEP;
BEGIN
    WRITE(CHR(7));
END;
```

```
PROCEDURE SCREEN_MAIN;
BEGIN
    CLRSCR;
```

Line 73 Col 1 Insert Indent A:GOESINTO.PAS

```
GOTOXY(15,1);WRITELN('! **** MENU **** !');
GOTOXY(15,2);WRITELN('! CTRL-R : RUN !');
GOTOXY(15,3);WRITELN('! CTRL-L : LIST !');
GOTOXY(15,4);WRITELN('! CTRL-P : PRINT !');
GOTOXY(15,5);WRITELN('! CTRL-Q : QUIT !');
```

END;

```
BEGIN
    QUIT := FALSE;
    REPEAT
        SCREEN_MAIN;
        ASSIGN (NAMEIN,'A:NAME.DAT');
        ASSIGN (BOMFILE,'A:BOMFILE.DAT');
        ASSIGN (TFILE,'A:TFILE.DAT');
        ASSIGN (UFILE,'A:UFILE.DAT');
        RESET(NAMEIN);
        RESET(BOMFILE);
        IMAX := FILESIZE(NAMEIN);
        WHILE NOT KEYSTAT(CH) DO BEGIN END;
        IF CH IN COMADSET
            THEN CASE ORD(CH) OF
                18 : BEGIN
                    GOTOXY(10,3);WRITELN('>> Running ...');
                    REWRITE(TFILE);
```

Line 97 Col 1 Insert Indent A:GOESINTO.PAS

```

REWRITE(UFILE);
FOR I := 1 TO IMAX DO
BEGIN
  READ(NAMEIN,NAMEREC);
  RV[I] := NAMEREC.REC_NO;
  CV[I] := NAMEREC.CODE;
END;
FOR RI := 1 TO IMAX DO
FOR CI := 1 TO IMAX DO
BEGIN
  MM[RI,CI] := 0;
  IF RI = CI
  THEN WM[RI,CI] := 1
  ELSE WM[RI,CI] := 0
END;
I := 1;
WHILE NOT EOF(BOMFILE) DO
BEGIN
  READ (BOMFILE,BOMREC);
  WITH BOMREC DO
  BEGIN
    FCV[I] := FATHERCODE;
    SCV[I] := SONCODE;
    GIV[I] := GOESINTO;

```

Line 121 Col 1 Insert Indent A:GOESINTO.PAS

```

END;
FOR J := 1 TO IMAX DO
BEGIN
  IF CV[J] = FCV[I]
  THEN CI := RV[J];
  IF CV[J] = SCV[I]
  THEN RI := RV[J];
END;
MM[RI,CI] := GIV[I];
I := I+1;
END;
FOR CI := 1 TO IMAX DO
FOR RI := 1 TO IMAX DO
  WRITE (UFILE,MM[RI,CI]);
JMAX := FILESIZE(BOMFILE);
FOR CI := IMAX-1 DOWNT0 1 DO
BEGIN
  FOR I := CI TO IMAX DO WV[I] := 0;
  FOR RI := IMAX DOWNT0 CI+1 DO
  BEGIN
    IF NOT (MM[RI,CI] = 0)
    THEN BEGIN
      FOR I := IMAX DOWNT0 RI DO
        WV[I] := WV[I]+MM[RI,CI]*WM[I,RI];

```

Line 145 Col 1 Insert Indent A:GOESINTO.PAS

```

    END;
    FOR J := IMAX DOWNTO CI DO
      WM[J,CI] := WM[J,CI]+WV[J];
    END;
    WM[1,1] := 1;
    FOR CI := 1 TO IMAX DO
      FOR RI := CI TO IMAX DO
        BEGIN
          WRITE (TFILE,WM[RI,CI]);
          WRITELN('T(',RI,',',CI,') = ',WM[RI,CI]:6:3);
        END;
      WRITELN;WRITELN('TFILE Filesize : ',FILESIZE(TFILE));WRITELN;
    END;
12 : BEGIN
    RESET(TFILE);
    RESET(UFILE);
    FOR CI := 1 TO IMAX DO
      FOR RI := CI TO IMAX DO
        BEGIN
          READ(TFILE,C);
          WM[RI,CI] := C;
          WRITELN('T(',RI,',',CI,') = ',WM[RI,CI]:6:3);
        END;

```

Line 169 Col 1 Insert Indent A:GOESINTO.PAS

```

    END;
16 : BEGIN
    RESET(TFILE);
    RESET(UFILE);
    FOR CI := 1 TO IMAX DO
      FOR RI := 1 TO IMAX DO
        BEGIN
          WM[RI,CI] := 0;
          MM[RI,CI] := 0;
        END;
      FOR CI := 1 TO IMAX DO
        FOR RI := CI TO IMAX DO
          READ (TFILE,WM[RI,CI]);
        FOR CI := 1 TO IMAX DO
          FOR RI := 1 TO IMAX DO
            READ (UFILE,MM[RI,CI]);
          SCREEN_MAIN;GOTOXY(10,10);WRITELN(')) Printing ...');
          WRITELN(LST,' ':45,'MATRIX');WRITELN(LST);
          WRITELN(LST);WRITE(LST,'CODE':10,' ':10);
          FOR I := 1 TO IMAX DO
            BEGIN
              READ(NAMEIN,NAMEREC);
              CV[I] := NAMEREC.CODE;
              WRITE(LST,CV[I]:9);

```

```

Line 193 Col 1 Insert Indent A:GOESINTO.PAS
  END;
  WRITELN(LST);WRITE(LST,'----':10,' ':10);
  FOR I := 1 TO IMAX DO WRITE(LST,'----':9);WRITELN(LST);
  FOR RI := 1 TO IMAX DO
  BEGIN
    WRITELN(LST);WRITELN(LST);WRITE(LST,CV[RI]:10,' ':10);
    FOR CI := 1 TO IMAX DO
      WRITE(LST,MM[RI,CI]:9:3);
    END;
  RESET(NAMEIN);
  WRITELN(LST,' ':45,'UMATRIX');WRITELN(LST);
  WRITELN(LST);WRITE(LST,'CODE':10,' ':10);
  FOR I := 1 TO IMAX DO
  BEGIN
    READ(NAMEIN,NAMEREC);
    CV[I] := NAMEREC.CODE;
    WRITE(LST,CV[I]:9);
  END;
  WRITELN(LST);WRITE(LST,'----':10,' ':10);
  FOR I := 1 TO IMAX DO WRITE(LST,'----':9);WRITELN(LST);
  FOR RI := 1 TO IMAX DO
  BEGIN
    WRITELN(LST);WRITELN(LST);WRITE(LST,CV[RI]:10,' ':10);
    FOR CI := 1 TO IMAX DO

```

```

Line 239 Col 1 Insert Indent A:GOESINTO.PAS

```

```

  WRITE(LST,MM[RI,CI]:9:3);
  END;
  WRITELN(LST);SCREEN_MAIN;
  END;
17 : QUIT := TRUE;
  END
  ELSE BEEP;
  UNTIL QUIT;
  CLOSE (NAMEIN);
  CLOSE (BOMFILE);
  CLOSE (TFILE);
  CLOSE (UFILE);
END.

```

Logged drive: A
Active directory: \

Work file: A:GOESINTO.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 7045 bytes
Free: 55590 bytes

)

Compiling
 230 lines

Code: 010A paragraphs (4256 bytes), 0C73 paragraphs free
Data: 0161 paragraphs (5648 bytes), 0E86 paragraphs free
Stack/Heap: 2FFD paragraphs (196560 bytes)

)

Line 1 Col 1 Insert Indent A:MRP.PAS
PROGRAM MATERIALS_REQUIREMENT_PLANNING;
TYPE

```

INDEX      = 1..20;
PERINDEX  = 1..20;
STRING4   = STRING[4];
STRING20  = STRING[20];
NAMERECORD = RECORD
            REC_NO : INTEGER;
            CODE   : STRING4;
            NAME   : STRING20;
            UNIT   : STRING4;
            CLASS  : STRING4;
            NAMELENGTH : INTEGER;
            END;
ORDERRECORD = RECORD
            CODE      : STRING4;
            ORDQUANT  : REAL;
            LEADTIME  : INTEGER;
            SSTOCK    : REAL;
            END;
PERNAMERECORD= RECORD
            REC_NO    : INTEGER;
            PERNAME   : STRING20;
            NAMELENGTH : INTEGER;

```

Line 25 Col 1 Insert Indent A:MRP.PAS
END;

```

RECNOVECTOR = ARRAY [INDEX] OF INTEGER;
CODEVECTOR  = ARRAY [INDEX] OF STRING4;
NAMEVECTOR  = ARRAY [INDEX] OF STRING20;
CLASSVECTOR = ARRAY [INDEX] OF STRING4;
UNITVECTOR  = ARRAY [INDEX] OF STRING4;
LENGTHVECTOR = ARRAY [INDEX] OF INTEGER;
PERNAMEVECTOR = ARRAY [INDEX] OF STRING20;
EQVECTOR    = ARRAY [INDEX] OF REAL;
LEADTIMEVECTOR = ARRAY [INDEX] OF INTEGER;
SAFETYSTOCKVECTOR = ARRAY [INDEX] OF REAL;
THATRIX     = ARRAY [INDEX,INDEX] OF REAL;
UMATRIX     = ARRAY [INDEX,INDEX] OF REAL;
DEMANDVECTOR = ARRAY [INDEX,PERINDEX] OF REAL;
PRODUCTIONVECTOR = ARRAY [INDEX,PERINDEX] OF REAL;
STORAGEVECTOR = ARRAY [INDEX] OF REAL;
ONHANDSTOCKVECTOR = ARRAY [INDEX] OF REAL;
ENDSTOCKVECTOR = ARRAY [INDEX] OF REAL;
GRSREQMATRIX = ARRAY [INDEX,PERINDEX] OF REAL;
NETREQMATRIX = ARRAY [INDEX,PERINDEX] OF REAL;
PORMATRIX    = ARRAY [INDEX,PERINDEX] OF REAL;
PORLMATRIX   = ARRAY [INDEX,PERINDEX] OF REAL;

```


Line 49 Col 1 Insert Indent A:MRP.PAS
VAR

```

QUIT      : BOOLEAN;
MAXITEM   : INTEGER;
MAXPERIOD : INTEGER;
I, RI, CI : INDEX;
PI, LI    : PERINDEX;
CH        : CHAR;
N         : INTEGER;
RV        : RECVVECTOR;
CV        : CODEVECTOR;
NV        : NAMEVECTOR;
KV        : CLASSVECTOR;
BV        : UNITVECTOR;
NL        : LENGTHVECTOR;
EQQ       : EQQVECTOR;
LTV       : LEADTIMEVECTOR;
SS        : SAFETYSTOCKVECTOR;
T         : TMATRIX;
U         : UMATRIX;
D         : DEMANDVECTOR;
P         : PRODUCTIONVECTOR;
OS        : ONHANDSTOCKVECTOR;
ES        : ENDSTOCKVECTOR;
MRP       : PORLMATRIX;

```

Line 73 Col 1 Insert Indent A:MRP.PAS

```

PNV       : PERNAMEVECTOR;
POR       : PORMATRIX;
PORL      : PORLMATRIX;
GRSREQ    : GRSREQMATRIX;
NETREQ    : NETREQMATRIX;
ORDERREC  : ORDERRECORD;
NAMEREC   : NAMERECORD;
PERNAMEREC : PERNAMERECORD;
INVOUT    : FILE OF ORDERRECORD;
NAMEIN    : FILE OF NAMERECORD;
PERNAMEIN : FILE OF PERNAMERECORD;
DFILE,PFILE,SFILE,OSFILE,ESFILE : FILE OF REAL;
TFILE,UFILE,GRQFILE,NREQFILE,MRPFILE : FILE OF REAL;

```

CONST

```

COMMANDSET : SET OF CHAR = [#7,#14,#16,#17,#18,#12];

```

FUNCTION KEYSTAT (VAR CH : CHAR) : BOOLEAN;

TYPE REG_PACK = RECORD

```

  AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
  END;

```

VAR REGISTERS : REG_PACK;

BEGIN

```

  REGISTERS.AX := $0600;

```

```

Line 97 Col 1 Insert Indent A:MRP.PAS
REGISTERS.DX := 255;
MSDOS (REGISTERS);
CH := CHR (REGISTERS.AX);
KEYSTAT := CH() CHR(0)
END;

PROCEDURE BEEP;
BEGIN
WRITE (CHR(7));
END;

PROCEDURE SCREEN_MAIN;
BEGIN
CLRSCR;
GOTOXY (12,1); WRITELN ('! ***** MENU ***** !');
GOTOXY (12,2); WRITELN ('! CTRL-R : RUN !');
GOTOXY (12,3); WRITELN ('! CTRL-L : LIST !');
GOTOXY (12,4); WRITELN ('! CTRL-P : PRINT !');
GOTOXY (12,5); WRITELN ('! CTRL-Q : QUIT !');
END;

Line 122 Col 1 Insert Indent A:MRP.PAS
PROCEDURE SCREEN_LIST;
BEGIN
CLRSCR;
GOTOXY (12,1); WRITELN ('! ***** MENU ***** !');
GOTOXY (12,2); WRITELN ('! CTRL-G : GROSS REQUIREMENTS !');
GOTOXY (12,3); WRITELN ('! CTRL-N : NET REQUIREMENTS !');
GOTOXY (12,4); WRITELN ('! CTRL-R : ORDER RELEASES !');
END;

BEGIN
QUIT := FALSE;
SCREEN_MAIN;
ASSIGN (NAMEIN, 'A:NAME.DAT');
ASSIGN (PERNAMEIN, 'A:PERNAME.DAT');
ASSIGN (INVOUT, 'A:INVOUT.DAT');
ASSIGN (TFILE, 'A:TFILE.DAT');
ASSIGN (UFILE, 'A:UFILE.DAT');
ASSIGN (DFILE, 'A:DFILE.DAT');
ASSIGN (PFILE, 'A:PFILE.DAT');
ASSIGN (SFILE, 'A:SFILE.DAT');
ASSIGN (OSFILE, 'A:OSFILE.DAT');
ASSIGN (ESFILE, 'A:ESFILE.DAT');
ASSIGN (GREQFILE, 'A:GREQFILE.DAT');

```

Line 146 Col 1 Insert Indent A:MRP.PAS

```

ASSIGN (NREQFILE,'A:NREQFILE.DAT');
ASSIGN (MRPFILE,'A:MRPFILE.DAT');
RESET (NAMEIN);
RESET (PERNAMEIN);
RESET (INVOUT);
RESET (TFILE);
RESET (UFILE);
RESET (DFILE);
RESET (PFILE);
RESET (SFILE);
RESET (OSFILE);
MAXITEM := FILESIZE(NAMEIN);
MAXPERIOD := FILESIZE(PERNAMEIN);
FOR I := 1 TO MAXITEM DO
BEGIN
  READ (NAMEIN,NAMEREC);
  WITH NAMEREC DO
  BEGIN
    RV[I] := REC_NO;
    CV[I] := CODE;
    NV[I] := NAME;
    BV[I] := UNIT;
    KV[I] := CLASS;
    NL[I] := NAMELENGTH;
  
```

Line 170 Col 1 Insert Indent A:MRP.PAS

```

END;
END;
FOR PI := 1 TO MAXPERIOD DO
BEGIN
  READ (PERNAMEIN,PERNAMEREC);
  PNV[PI] := PERNAMEREC.PERNAME;
END;
WHILE NOT QUIT DO
BEGIN
  REPEAT
    WHILE NOT KEYSTAT (CH) DO BEGIN END;
    IF NOT (CH IN COMMANDSET) THEN BEEP;
  UNTIL (CH IN COMMANDSET);
  IF (ORD (CH) = 17) THEN QUIT := TRUE;
  CASE ORD(CH) OF
  18 : BEGIN
    GOTOXY(6,8); WRITELN(') Running ... ');
    REWRITE (ESFILE);
    REWRITE (GREQFILE);
    REWRITE (NREQFILE);
    REWRITE (MRPFILE);
  
```

Line 189 Col 1 Insert Indent A:MRP.PAS

```

FOR PI := 1 TO MAXPERIOD DO
  FOR I := 1 TO MAXITEM DO
    BEGIN
      READ (DFILE,D[I,PI]);
      READ (PFILE,P[I,PI]);
    END;
  FOR CI := 1 TO MAXITEM DO
    FOR RI := CI TO MAXITEM DO
      READ (TFILE,T[RI,CI]);
    FOR CI := 1 TO MAXITEM DO
      FOR RI := 1 TO MAXITEM DO
        READ (UFILE,U[RI,CI]);
      FOR I := 1 TO MAXITEM DO
        BEGIN
          READ (OSFILE,OS[I]);
          READ (INVOUT,ORDERREC);
          WITH ORDERREC DO
            BEGIN
              EQQ[I] := ORDQUANT;
              LTV[I] := LEADTIME;
              SS[I] := SSTOCK;
            END;
          END;
        FOR RI := 1 TO MAXITEM DO

```

Line 213 Col 1 Insert Indent A:MRP.PAS

```

FOR CI := 1 TO MAXPERIOD DO
  BEGIN
    GRSREQ[RI,CI] := 0;
    NETREQ[RI,CI] := 0;
    POR[RI,CI] := 0;
    PORL[RI,CI] := 0;
  END;
  FOR RI := 1 TO MAXITEM DO
    FOR PI := 1 TO MAXPERIOD DO
      FOR I := 1 TO MAXITEM DO
        NETREQ[RI,PI] := NETREQ[RI,PI]+T[RI,I]*D[I,PI];
      FOR I := 1 TO MAXITEM DO
        FOR PI := 1 TO MAXPERIOD DO
          WRITE (NREQFILE,NETREQ[I,PI]);
        FOR I := 1 TO MAXITEM DO
          BEGIN
            OS[I] := OS[I] - SS[I];
            FOR PI := 1 TO MAXPERIOD DO
              BEGIN
                GRSREQ[I,PI] := D[I,PI] - P[I,PI] - OS[I];
                IF (GRSREQ[I,PI] <= 0)
                  THEN BEGIN
                    OS[I] := - GRSREQ[I,PI];
                    GRSREQ[I,PI] := 0;

```

Line 237 Col 1 Insert Indent A:MRP.PAS

```

END
ELSE BEGIN
  IF (EQQ[I] <> 0)
    THEN REPEAT
      POR[I,PI] := POR [I,PI] + EQQ[I];
      UNTIL (POR[I,PI] >= GRSREQ[I,PI])
    ELSE POR[I,PI] := GRSREQ[I,PI];
    OS[I] := POR[I,PI] - GRSREQ[I,PI];
    LI := PI-LTV[I];
    PORL[I,LI] := POR [I,PI];
    FOR RI := I+1 TO MAXITEM DO
      D[RI,LI] := D[RI,LI]+PORL[I,LI]*U[RI,I];
    END;
    WRITE (GREQFILE,D[I,PI]);
  END;
  WRITE (ESFILE,OS[I]);
  END;
FOR RI := 1 TO MAXITEM DO
  FOR CI := 1 TO MAXPERIOD DO
    WRITE (MRPFILE,PORL[RI,CI]);
  SCREEN_MAIN;
  GOTOXY(6,8); WRITELN('>> Run Completed. ');
END;
12 : BEGIN

```

Line 261 Col 1 Insert Indent A:MRP.PAS

```

SCREEN_LIST;
WHILE NOT KEYSTAT(CH) DO BEGIN END;
CASE ORD(CH) OF
14: BEGIN
  RESET(NREQFILE);
  CLRSCR;WRITELN (' ':15,'NET REQUIREMENTS');WRITELN;
  FOR I := 1 TO MAXITEM DO
    FOR PI := 1 TO MAXPERIOD DO
      BEGIN
        READ (NREQFILE,NETREQ[I,PI]);
        WRITELN ('NETREQ ('I','PI') =',NETREQ[I,PI]);
      END;
  WRITELN ('>> Press any key to continue ...');
  WHILE NOT KEYSTAT(CH) DO BEGIN END;
  SCREEN_MAIN;
END;
7: BEGIN
  CLRSCR;WRITELN (' ':15,'GROSS REQUIREMENTS');WRITELN;
  RESET (GREQFILE);
  FOR I := 1 TO MAXITEM DO
    FOR PI := 1 TO MAXPERIOD DO
      BEGIN
        READ (GREQFILE,GRSREQ[I,PI]);
        WRITELN ('GRSREQ ('I','PI') =',GRSREQ[I,PI]);
      END;
  END;

```

Line 331 Col 1 Insert Indent A:MRP.PAS

```

END;
16: BEGIN
  SCREEN_LIST;
  WHILE NOT KEYSTAT(CH) DO BEGIN END;
  CASE ORD(CH) OF
14: BEGIN
    GOTOXY (6,8);WRITELN (')) Printing ...');WRITELN(LST);
    RESET (NREQFILE);
    FOR RI := 1 TO MAXITEM DO
      FOR PI := 1 TO MAXPERIOD DO
        READ (NREQFILE,NETREQ[RI,PI]);
        WRITELN (LST,' ':60,'NET REQUIREMENT AMOUNTS');
        WRITELN (LST,' ':60,'-----');
        WRITELN(LST);WRITELN(LST);
        WRITELN (LST,' ':65,'NET AMOUNTS REQUIRED ');
        WRITE (LST,'NO':10,'CODE':10,'NAME':14,'UNIT':16,'CLASS':8);
        FOR PI := 1 TO MAXPERIOD DO
          WRITE (LST,PNV[PI]:13);
        WRITELN(LST);
        WRITE(LST,'--':10,'----':10,'-----':14,'-----':16,'-----':8);
        FOR PI := 1 TO MAXPERIOD DO
          WRITE (LST,'-----':13);
        FOR I := 1 TO MAXITEM DO
          BEGIN

```

Line 355 Col 1 Insert Indent A:MRP.PAS

```

  WRITELN(LST);WRITELN(LST);
  WRITE(LST,I:10,CV[I]:10,' ':6,NV[I]:NL[I],' ':20-NL[I],
    BV[I]:4,KV[I]:6,' ':2);
  FOR PI := 1 TO MAXPERIOD DO
    WRITE(LST,NETREQ[I,PI]:13:3);
  END;
  WRITELN(LST);SCREEN_MAIN;
  END;
7: BEGIN
  GOTOXY (6,8);WRITELN (')) Printing ...');WRITELN(LST);
  RESET (GREQFILE);
  FOR RI := 1 TO MAXITEM DO
    FOR PI := 1 TO MAXPERIOD DO
      READ (GREQFILE,GRSREQ[RI,PI]);
      WRITELN (LST,' ':60,'GROSS REQUIREMENT AMOUNTS');
      WRITELN (LST,' ':60,'-----');
      WRITELN(LST);WRITELN(LST);
      WRITELN (LST,' ':65,'GROSS AMOUNTS REQUIRED ');
      WRITE (LST,'NO':10,'CODE':10,'NAME':14,'UNIT':16,'CLASS':8);
      FOR PI := 1 TO MAXPERIOD DO
        WRITE (LST,PNV[PI]:13);
      WRITELN(LST);
      WRITE(LST,'--':10,'----':10,'-----':14,'-----':16,'-----':8);
      FOR PI := 1 TO MAXPERIOD DO

```

```

Line 379 Col 1 Insert Indent A:MRP.PAS
      WRITE (LST,'-----':13);
FOR I := 1 TO MAXITEM DO
BEGIN
  WRITELN(LST);WRITELN(LST);
  WRITE(LST,I:10,CV[I]:10,' ':6,NV[I]:NL[I],' ':20-NL[I],
        BV[I]:4,KV[I]:6,' ':2);
  FOR PI := 1 TO MAXPERIOD DO
    WRITE(LST,GRSREQ[I,PI]:13:3);
  END;
  WRITELN(LST);SCREEN_MAIN;
END;
18: BEGIN
  GOTOXY (6,8);WRITELN ('') Printing ...');WRITELN(LST);
  RESET (MRPFILE);
  RESET (ESFILE);
  FOR RI := 1 TO MAXITEM DO
    FOR CI := 1 TO MAXPERIOD DO
      READ (MRPFILE,PORL[RI,CI]);
  FOR RI := 1 TO MAXITEM DO
    READ (ESFILE,OS[RI]);
  WRITELN (LST,' ':60,'MATERIAL REQUIREMENTS PLANNING');
  WRITELN (LST,' ':60,'-----');
  WRITELN(LST);WRITELN(LST);
  WRITELN (LST,' ':65,'AMOUNT TO BE PRODUCED/ORDERED');

Line 403 Col 1 Insert Indent A:MRP.PAS
      WRITE (LST,'NO':10,'CODE':10,'NAME':14,'UNIT':16,'CLASS':8);
FOR PI := 1 TO MAXPERIOD DO
  WRITE (LST,PNV[PI]:13);
WRITE(LST,'AMOUNT IN STOCK':20);WRITELN(LST);
WRITE(LST,'--':10,'----':10,'----':14,'----':16,'----':8);
FOR PI := 1 TO MAXPERIOD DO
  WRITE (LST,'-----':13);
WRITE(LST,'-----':20);WRITELN(LST);
FOR I := 1 TO MAXITEM DO
BEGIN
  WRITELN(LST);WRITELN(LST);
  WRITE(LST,I:10,CV[I]:10,' ':6,NV[I]:NL[I],' ':20-NL[I],
        BV[I]:4,KV[I]:6,' ':2);
  FOR PI := 1 TO MAXPERIOD DO
    WRITE(LST,PORL[I,PI]:13:3);
  WRITE(LST,OS[I]:15:3);
  END;
  WRITELN(LST);SCREEN_MAIN;
END;
END;
END;
END;
CLOSE (NAMEIN);

```

```
Line 449 Col 1 Insert Indent A:MRP.PAS  
CLOSE (PERNAMEIN);  
CLOSE (INVOUT);  
CLOSE (TFILE);  
CLOSE (UFILE);  
CLOSE (DFILE);  
CLOSE (PFILE);  
CLOSE (SFILE);  
CLOSE (OSFILE);  
CLOSE (ESFILE);  
CLOSE (GREQFILE);  
CLOSE (NREQFILE);  
CLOSE (MRPFILE);  
END.
```


Logged drive: A
Active directory: \

Work file: A:MRP.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 16118 bytes
Free: 46517 bytes

)

Compiling
 439 lines

Code: 0268 paragraphs (9904 bytes), 0812 paragraphs free
Data: 05E0 paragraphs (24064 bytes), 0A07 paragraphs free
Stack/Heap: 27E6 paragraphs (163424 bytes)

)

```

Line 1 Col 1 Insert Indent A:CRP.PAS
PROGRAM CAPACITY_REQUIREMENTS_PLANNING;
CONST

```

```

  N = 1;

```

```

  M = 1;

```

```

TYPE

```

```

  INDEX = 1..20;

```

```

  PERINDEX = 1..20;

```

```

  DEPTINDEX = 1..5;

```

```

  STRING4 = STRING[4];

```

```

  STRING20 = STRING[20];

```

```

  NAMERECORD = RECORD

```

```

    REC_NO : INTEGER;

```

```

    CODE : STRING4;

```

```

    NAME : STRING20;

```

```

    UNIT : STRING4;

```

```

    CLASS : STRING4;

```

```

    NAMELENGTH : INTEGER;

```

```

  END;

```

```

  PERNAMERECORD = RECORD

```

```

    REC_NO : INTEGER;

```

```

    PERNAME : STRING20;

```

```

    NAMELENGTH : INTEGER;

```

```

  END;

```

```

  DEPTNAMERECORD = RECORD

```

```

Line 25 Col 1 Insert Indent A:CRP.PAS

```

```

    REC_NO : INTEGER;

```

```

    DEPTNAME : STRING20;

```

```

    NAMELENGTH : INTEGER;

```

```

  END;

```

```

  RECNOVECTOR = ARRAY [INDEX] OF INTEGER;

```

```

  CODEVECTOR = ARRAY [INDEX] OF STRING4;

```

```

  NAMEVECTOR = ARRAY [INDEX] OF STRING20;

```

```

  CLASSVECTOR = ARRAY [INDEX] OF STRING4;

```

```

  UNITVECTOR = ARRAY [INDEX] OF STRING4;

```

```

  LENGTHVECTOR = ARRAY [INDEX] OF INTEGER;

```

```

  PERNAMEVECTOR = ARRAY [PERINDEX] OF STRING20;

```

```

  DEPTNAMEVECTOR = ARRAY [DEPTINDEX] OF STRING20;

```

```

  OPTINES = ARRAY [DEPTINDEX,INDEX] OF REAL;

```

```

  CAPACITY = ARRAY [DEPTINDEX,PERINDEX] OF REAL;

```

```

  PORLVECTOR = ARRAY [INDEX,PERINDEX] OF REAL;

```

```

  CRPMATRIX = ARRAY [DEPTINDEX,PERINDEX] OF REAL;

```

```

  EXCESSNOMINAL = ARRAY [DEPTINDEX,PERINDEX] OF REAL;

```

```

  EXCESSMAXIMUM = ARRAY [DEPTINDEX,PERINDEX] OF REAL;

```

```

  AVAILABLETIME = ARRAY [DEPTINDEX,PERINDEX] OF REAL;

```

```

VAR

```

```

  MAXITEM,MAXPERIOD,MAXDEPT : INTEGER;

```

```

  CH : CHAR;

```

```

  QUIT : BOOLEAN;

```

```

  I,RI,CI : INDEX;

```

Line 49 Col 1 Insert Indent A:CRP.PAS

```

PI : PERINDEX;
DI : DEPTINDEX;
RV : RECVVECTOR;
CV : CODEVECTOR;
NV : NAMEVECTOR;
KV : CLASSVECTOR;
BV : UNITVECTOR;
NL : LENGTHVECTOR;
PNV : PERNAMEVECTOR;
DNV : DEPTNAMEVECTOR;
ENM : EXCESSNOMINAL;
EMM : EXCESSMAXIMUM;
ATM : AVAILBLETIME;
C : CAPACITY;
O : OPTIMES;
PORL: PORLVECTOR;
CRP : CRPMATRIX;
NAMEREC : NAMERECORD;
PERNAMEREC : PERNAMERECORD;
DEPTNAMEREC : DEPTNAMERECORD;
NAMEIN : FILE OF NAMERECORD;
PERNAMEIN : FILE OF PERNAMERECORD;
DEPTNAMEIN : FILE OF DEPTNAMERECORD;
OFIL,OFIL,MRPFIL,CRPFIL : FILE OF REAL;

```

Line 73 Col 1 Insert Indent A:CRP.PAS

```

CONST
COMMANDSET : SET OF CHAR = [#12,#16,#17,#18];

```

```

FUNCTION KEYSTAT (VAR CH :CHAR) : BOOLEAN;
TYPE REG_PACK = RECORD
    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
    END;
VAR REGISTERS : REG_PACK;
BEGIN
    REGISTERS.AX := $0600;
    REGISTERS.DX := 255;
    MSDOS (REGISTERS);
    CH := CHR(REGISTERS.AX);
    KEYSTAT := CH<>CHR(0)
END;

```

```

PROCEDURE BEEP;
BEGIN
    WRITE (CHR(7));
END;

```

```

Line 99 Col 1 Insert Indent A:CRP.PAS
PROCEDURE SCREEN_MAIN;
BEGIN
  CLRSCR;
  GOTOXY (12,1); WRITELN ('!      **** MENU ****      !');
  GOTOXY (12,2); WRITELN ('!      CTRL-R : RUN      !');
  GOTOXY (12,3); WRITELN ('!      CTRL-L : LIST      !');
  GOTOXY (12,4); WRITELN ('!      CTRL-P : PRINT      !');
  GOTOXY (12,5); WRITELN ('!      CTRL-Q : QUIT      !');
END;

```

```

BEGIN
  QUIT := FALSE;
  SCREEN_MAIN;
  ASSIGN (NAMEIN, 'A:NAME.DAT');
  ASSIGN (PERNAMEIN, 'A:PERNAME.DAT');
  ASSIGN (DEPTNAMEIN, 'A:DEPTNAME.DAT');
  ASSIGN (OFILE, 'A:OFILE.DAT');
  ASSIGN (CFILE, 'A:CFILE.DAT');
  ASSIGN (MRPFILE, 'A:MRPFILE.DAT');
  ASSIGN (CRPFILE, 'A:CRPFILE.DAT');
  RESET (NAMEIN);
  RESET (PERNAMEIN);

```

```

Line 123 Col 1 Insert Indent A:CRP.PAS
RESET (DEPTNAMEIN);
RESET (OFILE);
RESET (CFILE);
RESET (MRPFILE);
MAXITEM := FILESIZE (NAMEIN);
MAXPERIOD := FILESIZE (PERNAMEIN);
MAXDEPT := FILESIZE (DEPTNAMEIN);
FOR I := 1 TO MAXITEM DO
  BEGIN
    READ (NAMEIN, NAMEREC);
    WITH NAMEREC DO
      BEGIN
        RV[I] := REC_NO;
        CV[I] := CODE;
        NV[I] := NAME;
        BV[I] := UNIT;
        KV[I] := CLASS;
        NL[I] := NAMELENGTH;
      END;
  END;
FOR PI := 1 TO MAXPERIOD DO
  BEGIN
    READ (PERNAMEIN, PERNAMEREC);
    PNV[PI] := PERNAMEREC.PERNAME;

```

```

Line 147 Col 1 Insert Indent A:CRP.PAS
END;
FOR DI := 1 TO MAXDEPT DO
BEGIN
  READ (DEPTNAMEIN,DEPTNAMEREC);
  DNV[DI] := DEPTNAMEREC.DEPTNAME;
END;
WHILE NOT QUIT DO
BEGIN
  REPEAT
  WHILE NOT KEYSTAT (CH) DO BEGIN END;
  IF NOT (CH IN COMMANDSET) THEN BEEP;
  UNTIL (CH IN COMMANDSET);
  CASE ORD(CH) OF
  17 : QUIT := TRUE;
  18 : BEGIN
    GOTOXY (6,8); WRITELN ('> Running ... ');
    REWRITE (CRPFILE);
    FOR DI := 1 TO MAXDEPT DO
      FOR I := 1 TO MAXITEM DO
        READ (OFILE,O[DI,I]);
    FOR PI := 1 TO MAXPERIOD DO
      FOR DI := 1 TO MAXDEPT DO
        READ (CFILE,C[DI,PI]);
    FOR RI := 1 TO MAXITEM DO

```

```

Line 171 Col 1 Insert Indent A:CRP.PAS
  FOR CI := 1 TO MAXPERIOD DO
    READ (MRPFILE,PORL[RI,CI]);
  FOR DI := 1 TO MAXDEPT DO
    FOR PI := 1 TO MAXPERIOD DO
      BEGIN
        CRP[DI,PI] := 0;
        ENM[DI,PI] := 0;
        EMM[DI,PI] := 0;
        ATM[DI,PI] := 0;
      END;
    FOR DI := 1 TO MAXDEPT DO
      FOR PI := 1 TO MAXPERIOD DO
        FOR RI := 1 TO MAXITEM DO
          CRP[DI,PI] := CRP[DI,PI]+O[DI,RI]*PORL[RI,PI];
        FOR DI := 1 TO MAXDEPT DO
          FOR PI := 1 TO MAXPERIOD DO
            IF CRP[DI,PI] > N*C[DI,PI]
              THEN BEGIN
                IF CRP[DI,PI] > N*C[DI,PI]
                  THEN BEGIN
                    ENM[DI,PI] := CRP[DI,PI]-N*C[DI,PI];
                    EMM[DI,PI] := EMM[DI,PI]+N*C[DI,PI];
                  END
                ELSE ENM[DI,PI] := CRP[DI,PI]-N*C[DI,PI];

```

```

Line 195 Col 1 Insert Indent A:CRP.PAS
      END
      ELSE ATM[DI,PI] := N*C[DI,PI]-CRP[DI,PI];
FOR DI := 1 TO MAXDEPT DO
  FOR PI := 1 TO MAXPERIOD DO
    WRITE (CRPFILE,CRP[DI,PI],ENM[DI,PI],ENM[DI,PI],ATM[DI,PI]);
  SCREEN_MAIN;
  GOTOXY (6,8); WRITELN ('>> Run Completed.');
```

12 : BEGIN

```

  RESET (CRPFILE);
  FOR DI := 1 TO MAXDEPT DO
    FOR PI := 1 TO MAXPERIOD DO
      READ (CRPFILE,CRP[DI,PI],ENM[DI,PI],ENM[DI,PI],ATM[DI,PI]);
      CLRSCR;GOTOXY(4,5);
    WRITELN (' :15,'CAPACITY REQUIREMENTS PLANNING');GOTOXY(7,5);
  FOR PI := 1 TO MAXPERIOD DO
    BEGIN
      GOTOXY(10,10);
      WRITELN ('PERIOD : ',PNV[PI], ' :10);WRITELN;WRITELN;
      FOR DI := 1 TO MAXDEPT DO
        BEGIN
          WRITELN (CRP [DI,PI]:10:1,' OPERATING TIME IS REQUIRED ',
            'AT ',DNV[DI],'.');
          IF NOT (ATM[DI,PI] > 0)
```

```

Line 219 Col 1 Insert Indent A:CRP.PAS
      THEN IF ENM[DI,PI] > ENM [DI,PI]
        THEN WRITELN (ENM[DI,PI]:10:1,
          ' TIME UNITS EXCEED THE MAX. ',
          'CAPACITY AT ',DNV[DI],'.')
        ELSE WRITELN (ENM[DI,PI]:10:1,
          ' TIME UNITS EXCEED THE NOM. ',
          'CAPACITY AT ',DNV[DI],'.')
      ELSE WRITELN (ATM[DI,PI]:10:1,' TIME UNITS IS ',
        'FURTHER AVAILABLE AT ',DNV[DI],'.');
      WRITELN;
      END;
      WRITELN;WRITELN;WRITE ('Press any key to continue ...');
      WHILE NOT KEYSTAT (CH) DO BEGIN END;
      END;
      IF PI = MAXPERIOD THEN SCREEN_MAIN
      END;
16 : BEGIN
  GOTOXY (6,8); WRITELN ('>> Printing ... ');
  RESET (CRPFILE);
  FOR DI := 1 TO MAXDEPT DO
    FOR PI := 1 TO MAXPERIOD DO
      READ (CRPFILE,CRP[DI,PI],ENM[DI,PI],ENM[DI,PI],ATM[DI,PI]);
    WRITELN (LST,' :60,'CAPACITY REQUIREMENTS PLANNING');
    WRITELN (LST,' :60,'-----');
```

Line 243 Col 1 Insert Indent A:CRP.PAS

```

FOR DI := 1 TO MAXDEPT DO
  BEGIN
    WRITELN(LST);WRITELN(LST);WRITELN(LST);WRITELN(LST);
    WRITELN(LST,'DEPT : ',DNV[DI]);
    WRITELN(LST);WRITE(LST,' ':35);
    FOR PI := 1 TO MAXPERIOD DO
      WRITE (LST,PNV[PI]:13);
    WRITELN(LST);WRITE(LST,' ':35);
    FOR PI := 1 TO MAXPERIOD DO
      WRITE (LST,'-----':13);
    WRITELN(LST);WRITELN(LST);
    WRITE(LST,'REQUIRED OPERATING TIME      ');
    FOR PI := 1 TO MAXPERIOD DO
      WRITE (LST,CRP[DI,PI]:13:3);
    WRITELN(LST);WRITELN(LST);
    WRITE(LST,'TIME EXCEEDING NOM. CAPACITY  ');
    FOR PI := 1 TO MAXPERIOD DO
      WRITE (LST,ENM[DI,PI]:13:3);
    WRITELN(LST);WRITELN(LST);
    WRITE(LST,'TIME EXCEEDING MAX. CAPACITY ','(,M,') ');
    FOR PI := 1 TO MAXPERIOD DO
      WRITE (LST,ENM[DI,PI]:13:3);
    WRITELN(LST);WRITELN(LST);
    WRITE(LST,'TIME FURTHER AVAILABLE      ');
  
```

Line 289 Col 1 Insert Indent A:CRP.PAS

```

FOR PI := 1 TO MAXPERIOD DO
  WRITE (LST,ATM[DI,PI]:13:3);
  WRITELN(LST);
  END;
  SCREEN_MAIN;
  END;
  END;

```

```

END;
CLOSE (NAMEIN);
CLOSE (PERNAMEIN);
CLOSE (DEPTNAMEIN);
CLOSE (OFILE);
CLOSE (CFILE);
CLOSE (MRPFILE);
CLOSE (CRPFILE);

```

END.

Logged drive: A
Active directory: \

Work file: A:CRP.PAS
Main file:

Edit Compile Run Save

Dir Quit compiler Options

Text: 9889 bytes
Free: 52746 bytes

)

Compiling
 282 lines

Code: 0178 paragraphs (6016 bytes), 0C05 paragraphs free
Data: 01E5 paragraphs (7760 bytes), 0E02 paragraphs free
Stack/Heap: 2E59 paragraphs (189840 bytes)

)

REFERENCES NOT CITED

1. Wight, O.W. Production and Inventory Management in the Computer Age. Boston, Cashners Publishing Co., Inc., 1974.
2. Burbidge, John L. The Principles of Production Control. London, Macdonald and Evans Ltd., 1971.
3. Gavett, J. William Production and Operations Management. Harcout, Brace and World Inc., 1968.
4. Olsen, Robert A. Manufacturing Management : A Quantitative Approach. Scranton, International Textbook Company, 1968.
5. Tatar, Tefvik işletmelerde Üretim Yönetimi ve Teknikleri. Ankara, Ankara Devlet Mühendislik ve Mimarlık Akedemisi Yayınları, No:4.
6. Johnson, L. A. and Montgomery, D. C. Operations Research in Production Planning, Scheduling and Inventory Control. John Wiley & Sons, Inc., 1974
7. Tuğcu, Seyhan 'Classnotes AD 352', 1984.
8. Dağlı C., Nişancı i., Benli S. Üretim Planlaması ve Kontrolü. Segem, Yayın No 83, Ankara, 1981
9. Document of the World Bank, Turkey Agricultural Sector Restructuring Loan. 1985
10. Documents of 'Seminar on Trends and Developments in the Fertilizer Industry'. Istanbul, 1985.