Global Journals Inc.

Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 13 Issue 3 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Development of Material Requirements Planning (MRP) Software with C Language

By Md. Saiful Islam, Md. Mahbubur Rahman, Ripon Kumar Saha & Abu Md. Saifuddoha

Khulna University of Engineering & Technology (KUET), Bangladesh

Abstract - Now a day's a number of manufacturing firms in developing countries do not practice affordable, efficient and user friendly inventory management tools which has been identified as a major cause of high inventory cost for adequate planning. This study focuses on the development of Material Requirements Planning (MRP) software with programming language C that can be used by the local industries for inventory management in a job shop manufacturing environment. An algorithm has been developed to understand the MRP processing logic. A manual method of calculation to solve MRP problem has also been shown. Evaluation tests of the software were carried out using various products ranging from those with simple structure of single product to complex structure. The software was shown to be user friendly and allow for easy data input and output to be saved and retrieved for future planning. The input process of the software has been shown step by step. The output of the program shows the time-phased requirements for assemblies, parts and raw materials as well as the missing deliveries and time required to meet the missing deliveries.

Keywords : MRP, MPS, BOM, lead time, lot-sizing, inventory, software, C language.

GJCST-C Classification : K.6.1

Development of Material Requirements Planning (MRP) Software with C Language

Md. Saiful Islam [α], Md. Mahbubur Rahman [σ], Ripon Kumar Saha [ρ] & Abu Md. Saifuddoha [Ѡ]

Abstract - Now a day's a number of manufacturing firms in developing countries do not practice affordable, efficient and user friendly inventory management tools which has been identified as a major cause of high inventory cost for adequate planning. This study focuses on the development of Material Requirements Planning (MRP) software with programming language C that can be used by the local industries for inventory management in a job shop manufacturing environment. An algorithm has been developed to understand the MRP processing logic. A manual method of calculation to solve MRP problem has also been shown. Evaluation tests of the software were carried out using various products ranging from those with simple structure of single product to complex structure. The software was shown to be user friendly and allow for easy data input and output to be saved and retrieved for future planning. The input process of the software has been shown step by step. The output of the program shows the time-phased requirements for assemblies, parts and raw materials as well as the missing deliveries and time required to meet the missing deliveries.

Keywords : MRP, MPS, BOM, lead time, lot-sizing, inventory, software, C language.

I. Introduction

Materials Requirements Planning (MRP) is a computer-based production planning and inventory control system which is concerned with both production scheduling and inventory control. It is a material control system that attempts to keep adequate inventory levels to assure that required materials are available when needed. The main purpose of MRP software is to facilitate the calculation of requirements of materials and timing.

Thus it is a technique for determining the quantity and timing for the acquisition of dependent demand items needed to satisfy master production schedule (MPS) requirements by converting three inputs, bill of material, inventory data and master production schedule into time-phased requirements for subassemblies, component parts and raw materials, working backward from the due date using lead times and other information to determine when and how much to order [1]. The major objectives of an MRP system are to 1) ensure the availability of materials, components, and products for planned production and for customer delivery 2) maintain the lowest possible level of inventory 3) plan manufacturing activities, delivery schedules, and purchasing activities [2].

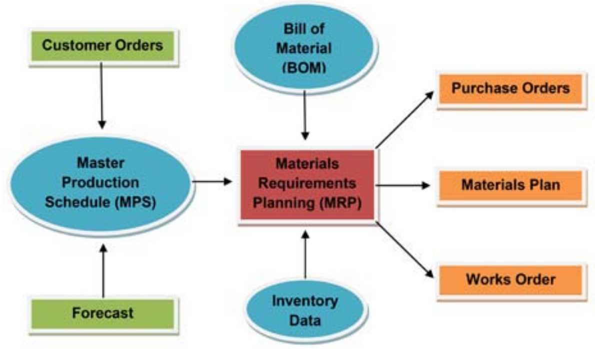


Figure 1 : Block diagram of MRP system

Before 1960's there was no satisfactory method available for handling the inventory of dependent items. A firm's formal inventory system was often patterned after order points and either misapplied or broken down into a maze of informal method when it comes to handling dependent items. There was no feasible method of keeping accurate records of thousands of inventory items to keep them out of too much scheduling trouble. Unfortunately, these did not always achieve the intended objective, however they always made a contribution to the inventory carrying and storage costs. During the 1960's the computer opened the door to an inventory system that could keep up to date records on the status of all inventory in stock. This brought a better understanding of production operation and new ways of managing production. It also brought out some new terminology, such as priority in material requirement planning system and capacity planning. The American production and inventory control society (APICS) has done much to standardize the terminology in this field and material requirement planning is popularly abbreviated as MRP in the world over [3].

In 1964, Joseph Orlicky as a response to the TOYOTA Manufacturing Program developed Material Requirements Planning (MRP). First company to use MRP was Black & Decker in 1964, with Dick Alban as project leader. In 1983 Oliver Wight developed MRP into manufacturing resource planning (MRP II) [4]. Orlicky's book is entitled The New Way of Life in Production and Inventory Management (1975). By 1975, MRP was implemented in 150 companies. This number had grown

Author α σ ρ Ѡ : Undergraduate student, Department of Industrial Engineering & Management (IEM), Khulna University of Engineering & Technology (KUET), Khulna-9203, Bangladesh.
E-mails : saifuliem@gmail.com, mahbub.iem@gmail.com, ripon_kuet27@yahoo.com, saifsf08@gmail.com

to about 8,000 by 1981. In the 1980s, Joe Orlicky's MRP evolved into Oliver Wight's manufacturing resource planning (MRP II) which brings master scheduling, rough-cut capacity planning, capacity requirements planning, S&OP in 1983 and other concepts to classical MRP. By 1989, about one third of the software industry was MRP II software sold to American industry ($1.2 billion worth of software) [5].

MRP is the way of life for many industries fabricating and assembling products like automobiles and radios. It is generally applicable in situations of multiple items with complex bills of materials and is especially suited to manufacturing settings where the demands of many of the components and subassemblies depend on the demands of items that face external demands. MRP is also suitable when the manufacturing cycle is long for the finished product and lead time for components and raw materials are relatively long. While demands for end items are independent the demands for components used to manufacture end items depend on the demands of the end items. The distinctions between independent and dependent demands are important in classifying inventory items and in developing systems to manage items within each demand classification [6].

The Master Production Schedule (MPS) includes quantities of products to be produced at a given time period. Quantities are included both at aggregate and detailed levels. Aggregate may refer to monthly production and detailed may refer to weekly or daily production. The master production schedule takes the form of a table in which rows represent products and columns represent time components [7]. The bill of materials (BOM) is a list of the raw materials, sub-assemblies, assemblies, parts and the quantities of each needed to manufacture an end product. Lead time is the time interval between ordering and receiving an item [8]. Sometimes it means the assembly time or processing time to produce an item.

A key variable in MRP system design is the selection of lot-sizing rule (how much to order) based on the lead time. The problem of lot sizing is one of satisfying the requirements while trying to minimize holding and setup costs. There are basically two major classes of lot sizing techniques namely Static and Dynamic. A static lot-sizing rule orders the same quantity each time an order is placed and often generating higher average on hand inventory for extra safety stock [9]. Dynamic decision rule changes the order quantity with each order such that each order is just large enough to prevent shortages over a specified time period by tying lot-size to gross requirements. It generally causes instability with lower-level components unable to respond sufficiently fast to changes in requirements [10] [11]. The lot-for-lot (LFL) ordering is the simplest approach and it refers to order the net requirements for a specific period. The LFL approach minimizes the holding cost by producing just-in-time. In lot-size (LS) ordering the order size must be in multiples of the lot size.

II. Methodology

The methodology used in developing the proposed MRP software adopts the following components namely, a) Problem Analysis b) Manual Method to Calculation c) Software Algorithm d) Software Input 6) Software Output.

a) Problem Analysis

To form a useful bill of material it is convenient to order the items by levels. The level of an item is the maximum number of stages of assembly required to get the item into an end product. Consider a specific final item named ‘A’ for a manufacturing firm.

MPS

Week/Day	3	7	8	10	12
Required Quantity (A)	250	200	150	300	250

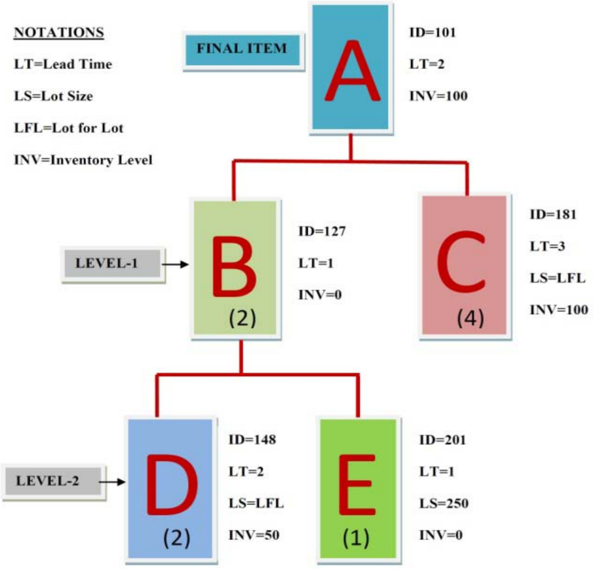


Figure 2 : MPS and Bill of Material for a specific product

According to above bill of material, one unit of final item ‘A’ requires two units of item ‘B’ and four units of item ‘C’. Again, one unit of item ‘B’ requires two units of item ‘D’ and one unit item ‘E’. Items ‘C’, ‘D’ and ‘E’ are purchased from different suppliers with various lead time. The beginning inventory levels of items ‘A’, ‘B’, ‘C’, ‘D’ and ‘E’ are 100, 0, 100, 50 and 0 respectively. The assembly time with items ‘B’ and ‘C’ to produce final item ‘A’ is 2 weeks. The assembly time with items ‘D’ and ‘E’ to produce item ‘B’ is 1 week. The time intervals between ordering and receiving for items ‘C’, ‘D’ and ‘E’ are 3, 2 and 1 week respectively. The items ‘C’ and ‘D’ are purchased with lot-for-lot (LFL) ordering policy but the item ‘E’ is purchased with lot-size (LS) ordering policy. There are total five orders in the master production schedule (MPS).

b) Manual Method of Calculation

MPS												
Week/Day	1	2	3	4	5	6	7	8	9	10	11	12
Required Quantity (A)			250				200	150		300		250
ITEM: A (LT= 2)												
Gross Requirements							200	150		300		250
Schedules Receipts												
Projected on Hand	100	100	100	100	100	100	100					
Net Requirements							100	150		300		250
Order Receipts							100	150		300		250
Order Releases					100	150		300		250		
ITEM: B (LT= 1)				100×2								
Gross Requirements					200	300		600		500		
Schedules Receipts												
Projected on Hand												
Net Requirements					200	300		600		500		
Order Receipts					200	300		600		500		
Order Releases				200	300		600		500			
ITEM: D (LT= 2)			200×2									
Gross Requirements				400	600		1200		1000			
Schedules Receipts												
Projected on Hand	50	50	50	50								
Net Requirements				350	600		1200		1000			
Order Receipts				350	600		1200		1000			
Order Releases		350	600		1200		1000					
ITEM: E (LT= 1)										500×1		
Gross Requirements				200	300		600		500			
Schedules Receipts												
Projected on Hand					50			150	150	150	150	150
Net Requirements				200	250		600		350			
Order Receipts				250	250		750		500			
Order Releases			250	250		750		500				
ITEM: C (LT= 3)											250×4	
Gross Requirements					400	600		1200		1000		
Schedules Receipts												
Projected on Hand	100	100	100	100	100							
Net Requirements					300	600		1200		1000		
Order Receipts					300	600		1200		1000		
Order Releases		300	600		1200		1000					

Table 1 : Manual method of calculating MRP problem

The required quantity of final item ‘A’ is 250 for the first order. The firm already has 100 units of final item in inventory. So the firm needs 150 units of ‘A’ to meet the first order. For 150 units of ‘A’ it requires 300 units of item ‘B’ and 600 units of item ‘C’. Again, for 300 units of ‘B’ the firm needs 600 units of item ‘D’ and 300 units of item ‘E’. The inventory level of item ‘D’ is 50. So, the net requirement of item ‘D’ is 550. If the firm orders today for 550 units of ‘D’ from supplier it can receive the order after two weeks. In the meantime the required quantity of item ‘E’ will be available for lower lead time. It will takes another one day to produce the required quantity of item ‘B’ with items ‘D’ and ‘E’. In the meantime the required quantity of item ‘C’ will be available. Finally, it will take another two more days to produce the required quantity of final item ‘A’ with items ‘B’ and ‘C’. It will take total five weeks to meet the first order. The first order can be delivered in the sixth week.

So, the order no.1 can't be met in third week. This order needs another 3 more weeks to meet.

c) Software Algorithm

Step 1 : Structure has been used to declare the variables of each item (item ID, item name, inventory level, lead time, lot size, number of item needed for each upper item, upper item ID, Number of lower item etc).

Step 2 : Input the total number of level and information of each item (Bill of Materials).

Step 3 : Input the total number of deliveries and timing & required quantity of final item for each delivery (Master Production Schedule).

Step 4 : Temporary inventory level variable is used to hold the inventory level of each item temporarily which will be destructed after calculation.

Step 5 : Finding out the order release date and temporary inventory level of independent item (final item) for each delivery

Step 6 : Finding out the order release date and temporary inventory level of all dependent items (sub items) for each delivery. If an item occurs more than once it will update the temporary inventory level for all repeated items simultaneously.

Step 7 : If the order release date of an item is negative it means that the Order Release falls before the planning period. So, the delivery can't be possible.

Step 8 : Then searching for the maximum negative value of order release date among all items for the deliveries which are not possible. This maximum negative value yields the number of weeks/days required to meet the missing deliveries.

Step 9 : For a missing delivery shift the inventory level of all items to the next delivery.

Step 10 : Again finding out the order release date and temporary inventory level of all items for the remaining deliveries because of missing the delivery

Step 11 : Repeat steps 9 and 10 for the remaining missing deliveries.

Step 12 : Holding the missing deliveries in an array.

Step 13 : Finally calculate the order release date and inventory level of all items for which deliveries are possible.

Step 14 : Show which deliveries can't be met and the required weeks/days to meet those deliveries.

Step 15 : Show the output of MRP that is the time-phased requirements for assemblies, parts and raw materials.

Step 16 : Show the final inventory level of all items.

Step 17 : Finally save the MRP output to a text file.

d) Software Input

```
Enter the total number of levels:  2
Enter the total number of items in level-1:  2
Enter the items ID for levels-1:
127
181
Enter the total number of items in level-2:  2
Enter the items ID for levels-2:
148
201
Give the information for final item:
-------------------------------------------
ID:  101
Name:  A
Inventory level:  100
Lead time:  2
Give the information for item ID-127:
----------------------------------------------
Name:  B
Upper item ID:  101
Number of lower item in the next level:  2
Number of item needed for each upper item:  2
Inventory level:  0
Lead time: 1
Give the information for item ID-181:
--------------------------------------------
Name:  C
Upper item ID:  101
Number of lower item in the next level:  0
Number of item needed for each upper item:  4
Inventory level:  100
Lead time:  3
Lot size [Enter 32000 when 'lot for lot' otherwise enter lot
size]:  32000
Give the information for item ID-148:
--------------------------------------------
Name:  D
Upper item ID:  127
Number of lower item in the next level:  0
Number of item needed for each upper item:  2
Inventory level:  50
Lead time: 2
Lot size [Enter 32000 when 'lot for lot' otherwise enter lot
size]:  32000
Give the information for item ID-201:
--------------------------------------------
Name:  E
Upper item ID:  127
Number of lower item in the next level:  0
Number of item needed for each upper item:  1
Inventory level:  0
Lead time: 1
Lot size [Enter 32000 when 'lot for lot' otherwise enter lot
size]:  250
Enter the total number of orders:  5
In which week/day order no.1 has to be released:  3
Number of final item required for order no.1:  250
In which week/day order no.2 has to be released:  7
Number of final item required for order no.1:  200
In which week/day order no.3 has to be released:  8
Number of final item required for order no.1:  150
In which week/day order no.4 has to be released:  10
Number of final item required for order no.1:  300
```

■ Command

🟦 Input

```
In which week/day order no.5 has to be released:  12
Number of final item required for order no.1:  250
```

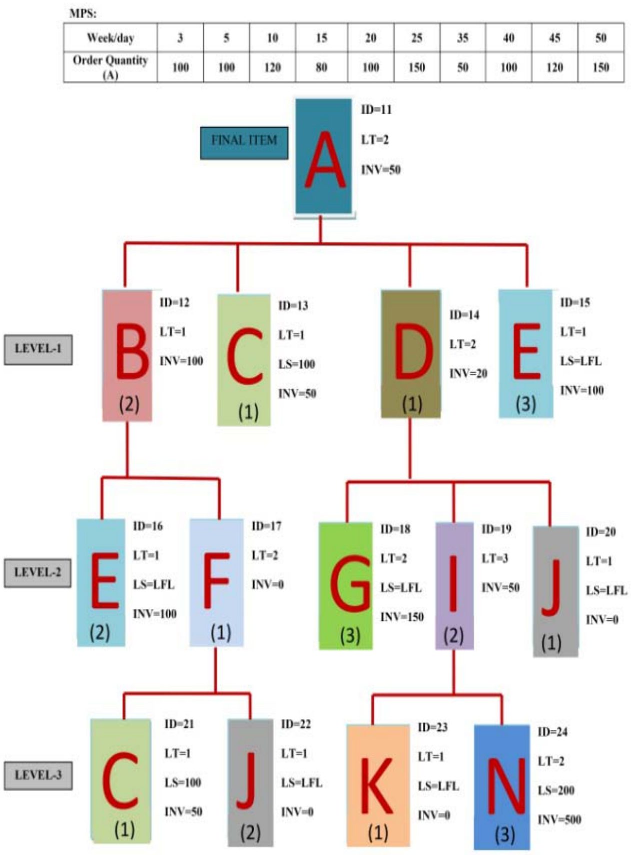
e) Software Output

```
        The order no.1 can't be met. This order needs
another 3 more days/weeks

            Item Name       Item ID            Quantity
Week/day-2
                C              181                 300
                D              148                 350
Week/day-3
                E              201                 250
                C              181                 600
                D              148                 600
Week/day-4
                B              127                 200
                E              201                 250
Week/day-5
                A              101                 100
                B              127                 300
                C              181                 1200
                D              148                 1200
Week/day-6
                A              101                 150
                E              201                 750
Week/day-7
                B              127                 600
                C              181                 1000
                D              148                 1000
Week/day-8
                A              101                 300
                E              201                 500
Week/day-9
                B              127                 500
Week/day-10
                A              101                 250

INVENTORY LEVEL:
-------------------------
Item Name        Item ID          Inventory Level
   A               101                  0
   B               127                  0
   C               181                  0
   D               148                  0
   E               201                 150
```

The firm has to start purchase process from week no. 2 to meet the remaining orders. It should order 300 units of item 'C' and 350 units of item 'D' from suppliers in second week and so on. The firm should start assembly process with items 'D' and 'E' to produce 200 units of item 'B'.



MPS:

Week/day	3	5	10	15	20	25	35	40	45	50
Order Quantity (A)	100	100	120	80	100	150	50	100	120	150

Figure 4 : MRP problem with complex BOM that can be solved with this software

III. Limitations

a) The maximum number of delivery is fifteen in MPS.
b) Capable of holding the information of thirty items for a specific MRP problem.
c) Not suitable to deliver the sub-assembly parts.
d) Only two types of lot sizing techniques were analyzed.
e) When user inputs wrong information then he has to start from the beginning.

IV. Conclusion

There are several sources of uncertainty that we have ignored so far. These include uncertainty in the quantity demanded (forecast errors) and the quantity supplied (yield losses), and uncertainty in the timing of demand and the timing of supply (random lead times). The largest cost or disadvantage of any MRP system is the cost of purchasing or leasing a computer system to support the function. However, with increasing inventory and production cost, along with decreasing computation costs the MRP system is getting easier to justify. The software is user friendly and was tested with various types of products and gave accurate results when verified with the manual method of calculation. So, the software is universal. The time-phased requirements for assemblies, parts and raw materials and final inventory level can easily be identified with the software. With the software local small scale manufacturing industries can improve customer service with lower inventories and reduce overtime and idle time.

References Références Referencias

1. Oladokun, V.O., & Olaitan, O.A. (2012). *Development of a Materials Requirements Planning (MRP) Software*. The Pacific Journal of Science and Technology, 13: 351-357.
2. Waldner, & Jean-Baptiste. (1992). *CIM: Principles of Computer Integrated Manufacturing,* John Wiley & Sons Ltd.
3. Jain, K.C., & Aggarwal, L.N. (2011). *Production planning control and Industrial Management,* 582- 597.
4. Hopp, W.J., & Spearman, M.L. (2004). Commissioned Paper To Pull or Not to Pull: What Is the Question? *Manufacturing & Service Operations Management.*
5. IE. (1991). *Competition in manufacturing leads to MRP II,* 23 (July) 10-13.
6. Starr, K.M. (1996). *Operation Management: A Systems Approach*. Boyd & Fraser: Chicago, IL.
7. Moustakis, D.V. (2000). *Material Requirements Planning,* 3.
8. Stevenson, W.J. (2011). *Operations Management* (8th Edition). New York, NY: McGraw-Hill Inc., 489 & 578.
9. Steinberge, S.J., & Napier, H.A. (1980*). Optimal Multi-Level Lot Sizing for Requirements Planning Systems*. Management Sci., 26:1258-1271
10. Biggs, J.R. (1979). *Heuristic Lot-Sizing and Sequencing Rules in a Multistage Production-Inventory System.* Decision Science, 19:91-115.
11. Lynn, A.F. (2006). *Material Requirement Planning: Tinkertoy Lawn Mower Activity.* Teaching Brief, Decision Sciences Journal of Innovative Education, 4(1).

Software Coding

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main() {
FILE *fp;
fp=fopen("MRP.txt", "w"); //Open a file to write the output information
// Use structure in which the variables of each item has been declared
struct item {
int id; // Item ID
char name[15]; // Item name
int u_id; // Upper item's ID
int item_n; // Item needed for each upper item
int n_l_item; // Number of lower items
int l_time; // Lead time
int i_level; // Inventory level
int t_i_level[16]; // Temporary inventory level
int level_n; // Item for which level
int l_size; // Lot size
int o_release[16]; // Order release
int o_quantity[16]; // Order quantity
};
// Variables declaration section
clrscr();
struct item info[30];
int level,a=1,b[20],d,i,j,k,l,m,n,s,q,Q,p[50],c[16],g[16],x,y;
// Input total levels and information of each item
printf("\nEnter the total number of levels: ");
scanf("%d", &level);
b[0]=1;
for(i=1;i<=level;i++) {
printf("\nEnter the total number of items in level-%d: ",i);
scanf("%d", &n);
b[i]=n;
printf("\nEnter the items ID for level-%d:\n",i);
for(j=1;j<=n;j++) {
scanf("%d", &info[a].id);
info[a].level_n=i;
a++;
} }
info[0].level_n=0;
printf("\n\nGive the information for final item:\n-----------------------------------\n");
printf("ID: ");
scanf("%d", &info[0].id);
printf("\nName: ");
scanf("%s",&info[0].name);
info[0].n_l_item=b[1];
printf("\nInventory level: ");
scanf("%d", &info[0].i_level);
printf("\nLead time: ");
scanf("%d", &info[0].l_time);
if (info[0].n_l_item==0) {
printf("\nLot size [use a huge number i.e.; 32000 when 'lot for lot' order]: ");
scanf("%d", &info[0].l_size); //Enter 32000 only when lot for lot otherwise use the lot size
}
for(i=1;i<a;i++) {
printf("\n\n\nGive the information for item ID-%d:\n-----------------------------------",info[i].id);
printf("\nName: ");
scanf("%s",&info[i].name);
printf("\nUpper item ID: ");
scanf("%d", &info[i].u_id);
printf("\nNumber of lower item in the next level: ");
scanf("%d", &info[i].n_l_item);
printf("\nNumber of item needed for each upper item: ");
scanf("%d", &info[i].item_n);
printf("\nInventory level: ");
```

```
scanf("%d", &info[i].i_level);
printf("\nLead time: ");
scanf("%d", &info[i].l_time);
if (info[i].n_l_item==0) {
printf("\nLot size [use a huge number i.e.; 32000 when 'lot for lot' order]: ");
scanf("%d", &info[i].l_size); //Enter 32000 only when lot for lot otherwise use the lot size
}
else
info[i].l_size=32000;
}
// Input the information of MPS
printf("\n\n\nEnter the total number of orders: ");
scanf("%d", &d);
printf("\n");
for(i=1;i<=d;i++) {
printf("\nIn which week/day order no.%d has to be released: ",i);
scanf("%d", &c[i]);
printf("\nNumber of final item required for delivery no.%d: ", i);
scanf("%d", &g[i]);
printf("\n");
}
s=c[i-1]; // The total number of weeks/days used according to MPS
// Use the temporary inventory level variables to hold the inventory level of each item temporarily which will be destructed after calculation
for(i=0;i<a;i++)
info[i].t_i_level[0]=info[i].i_level;
// Finding out the order release date and inventory level of independent item (final item) for each delivery
for(i=1;i<=d;i++) {
info[0].o_release[i]=c[i]-info[0].l_time;
info[0].o_quantity[i]=g[i]-info[0].t_i_level[i-1];
if(info[0].o_quantity[i]<=0) {
info[0].o_quantity[i]=0;
info[0].t_i_level[i]=info[0].t_i_level[i-1]-g[i];
}
else
info[0].t_i_level[i]=0;
}
// Finding out the order release date and inventory level of all dependent items (sub items) for each delivery
for(i=1;i<a;i++) {
for(j=0;j<a;j++) {
if(info[i].u_id==info[j].id) {
for(k=1;k<=d;k++) {
info[i].o_release[k]=info[j].o_release[k] - info[i].l_time;
info[i].o_quantity[k]=info[j].o_quantity[k]*info[i].item_n - info[i].t_i_level[k-1];
if(info[i].o_quantity[k]<=0) {
info[i].o_quantity[k]=0;
info[i].t_i_level[k]=info[i].t_i_level[k-1]-
info[j].o_quantity[k]*info[i].item_n;
for(y=1;y<a;y++) {
if(y==i)
continue;
else {
if(strcmp(info[i].name,info[y].name)==0) //Finding out for repeated item
info[y].t_i_level[k]=info[i].t_i_level[k];
} } }
else
info[i].t_i_level[k]=0;
for(y=1;y<a;y++) {
if(y==i)
continue;
else {
if(strcmp(info[i].name,info[y].name)==0)
info[y].t_i_level[k]=info[i].t_i_level[k];
} } } } } }
j=1;
// Finding out which deliveries can't be met and the number of days required to meet the missing deliveries
for(k=1;k<=d;k++) {
m=1;
if(info[0].o_release[k]<=0 && g[k]>info[0].t_i_level[k-1] && info[0].n_l_item==0) {
m=info[0].o_release[k];
}
if(info[0].t_i_level[k-1]>=g[k])
continue;
for(i=1;i<a;i++) {
for(q=0;q<a;q++) {
if(info[i].u_id==info[q].id) {
if(info[i].o_release[k]<=0 && info[q].o_quantity[k]*info[i].item_n>info[i].t_i_level[k-1] && info[i].n_l_item==0) {
if(info[i].o_release[k]<m)
m=info[i].o_release[k];
} } } }
for(l=1;l<=level;l++) {
n=0;
for(i=1;i<a;i++) {
if(info[i].level_n==l) {
for(q=0;q<a;q++) {
if(info[i].u_id==info[q].id) {
if(info[q].o_quantity[k]*info[i].item_n<=info[i].t_i_level[k-1])
n++;
} } } }
if(n==b[l]) {
if(info[0].level_n==l-1) {
if(info[0].o_release[k]<m)
m=info[0].o_release[k];
}
for(i=1;i<a;i++) {
if(info[i].level_n==l-1 && info[i].n_l_item !=0) {
for(q=0;q<a;q++) {
if(info[i].u_id==info[q].id) {
if(info[q].o_quantity[k]*info[i].item_n>info[i].t_i_level[k-
1]) {
if(info[i].o_release[k]<m)
m=info[i].o_release[k];
}} } } } } }
// When the value of m is negative, it means the delivery
is not possible and m holds the days required to meet
the delivery
if(m<=0) {
//Shifting the inventory level of all items of the missing
delivery to the next delivery
for(q=0;q<a;q++) {
info[q].t_i_level[k]=info[q].t_i_level[k-1];
}
//Again finding out the order release date and inventory
level of all items for the remaining deliveries because of
missing the delivery
for(i=k+1;i<=d;i++) {
info[0].o_release[i]=c[i]-info[0].l_time;
info[0].o_quantity[i]=g[i]-info[0].t_i_level[i-1];
if(info[0].o_quantity[i]<=0) {
info[0].o_quantity[i]=0;
info[0].t_i_level[i]=info[0].t_i_level[i-1]-g[i];
}
else
info[0].t_i_level[i]=0;
}
for(i=1;i<a;i++) {
for(x=0;x<a;x++) {
if(info[i].u_id==info[x].id) {
for(q=k+1;q<=d;q++) {
info[i].o_release[q]=info[x].o_release[q] - info[i].l_time;
info[i].o_quantity[q]=info[x].o_quantity[q]*info[i].item_n
- info[i].t_i_level[q-1];
if(info[i].o_quantity[q]<=0) {
info[i].o_quantity[q]=0;
info[i].t_i_level[q]=info[i].t_i_level[q-1]-
info[x].o_quantity[q]*info[i].item_n;
for(y=1;y<a;y++) {
if(y==i)
continue;
else {
if(strcmp(info[i].name,info[y].name)==0)
info[y].t_i_level[k]=info[i].t_i_level[k];
} } }
else {
info[i].t_i_level[q]=0;
for(y=1;y<a;y++) {
if(y==i)
continue;
else {
if(strcmp(info[i].name,info[y].name)==0)
info[y].t_i_level[k]=info[i].t_i_level[k];
} } } } } } }
p[j]=k; // Holding the missing deliveries in an array
j++;
x=-(1)*m;
if(x==0) {
printf("\n\nThe order no. %d can't be met. This order
needs another %d more day/week",k,x+1);
fprintf(fp,"\nThe order no. %d can't be met. This order
needs another %d more day/week",k,x+1);
}
else {
printf("\n\nThe order no. %d can't be met. This order
needs another %d more days/weeks",k,x+1);
fprintf(fp,"\nThe order no. %d can't be met. This order
needs another %d more days/weeks",k,x+1);
}} }
// Now calculate the order release date and inventory
level of all items for which deliveries are possible
for(k=1;k<=d;k++) {
q=1;
for(m=1;m<j;m++) {
if(k==p[m]) {
q=0;
break;
} }
if(q==0)
continue;
else {
if(info[0].n_l_item==0) {
info[0].o_quantity[k]=g[k]-info[0].i_level;
if(info[0].o_quantity[k]<=0) {
info[0].o_quantity[k]=0;
info[0].i_level=info[0].i_level-g[k];
}
else     if(info[0].o_quantity[k]<=info[0].l_size     &&
info[0].l_size==32000)
info[0].i_level=0;
else {
for(l=1;l<=50;l++) {
if(info[0].o_quantity[k]<=info[0].l_size * l) {
Q=info[0].l_size * l;
info[0].i_level=Q - info[0].o_quantity[k];
info[0].o_quantity[k]=Q;
break;
} } } }
else {
info[0].o_quantity[k]=g[k]-info[0].i_level;
if(info[0].o_quantity[k]<=0) {
info[0].o_quantity[k]=0;
info[0].i_level=info[0].i_level-g[k];
}
else
info[0].i_level=0;
} } }
for(i=1;i<a;i++) {
for(n=0;n<a;n++) {
if(info[i].u_id==info[n].id) {
for(k=1;k<=d;k++) {
q=1;
for(m=1;m<j;m++) {
if(k==p[m]) {
q=0;
```

```
break;
} }
if(q==0)
continue;
else {
info[i].o_quantity[k]=info[n].o_quantity[k]*info[i].item_n
- info[i].i_level;
Q=info[i].o_quantity[k];
if(Q<=0) {
info[i].o_quantity[k]=0;
info[i].i_level=info[i].i_level-
info[n].o_quantity[k]*info[i].item_n;
for(y=1;y<a;y++) {
if(y==i)
continue;
else {
if(strcmp(info[i].name,info[y].name)==0)
info[y].i_level=info[i].i_level;
} } }
else if(Q<=info[i].l_size && info[i].l_size==32000) {
info[i].i_level=0;
for(y=1;y<a;y++) {
if(y==i)
continue;
else {
if(strcmp(info[i].name,info[y].name)==0)
info[y].i_level=info[i].i_level;
} } }
else {
for(l=1;l<=50;l++) {
if(Q<=info[i].l_size * l) {
Q=info[i].l_size * l;
info[i].i_level=Q - info[i].o_quantity[k];
for(y=1;y<a;y++) {
if(y==i)
continue;
else {
if(strcmp(info[i].name,info[y].name)==0)
info[y].i_level=info[i].i_level;
}
}
info[i].o_quantity[k]=Q;
break;
} } } }} } } }
//The output section
printf("\n\n\n\n                    Item_Name        Item_ID
Quantity\n");
printf("          ---------    -------     --------------");
fprintf(fp,"\n\n                    Item_Name        Item_ID
Quantity\n");
fprintf(fp,"          ---------    -------     --------------");
for(i=1;i<=s;i++) {
x=i;
for(k=1;k<=d;k++) {
q=1;
for(m=1;m<j;m++) {
if(k==p[m]) {
q=0;
break;
} }
if(q==0) {
continue;
}
else {
for(m=0;m<a;m++) {
if(info[m].o_release[k]==i && info[m].n_l_item==0 &&
info[m].o_quantity[k]!=0) {
if(x==i) {
printf("\n\nWeek/day-%d:\n",i);
fprintf(fp,"\n\nWeek/day-%d:\n",i);
x++;
}
printf("            ");
fprintf(fp,"            ");
printf("%s",info[m].name);
fprintf(fp,"%s",info[m].name);
printf("                                              %d
%d\n",info[m].id,info[m].o_quantity[k]);
fprintf(fp,"                                          %d
%d\n",info[m].id,info[m].o_quantity[k]);
} } } } }
printf("\n\n\n\n\n INVENTORY LEVEL:\n ---------------");
fprintf(fp,"\n\n\n\n\n INVENTORY LEVEL:\n ---------------");
printf("\n\n Item_Name Item_ID Inventory_level\n");
printf(" --------- ------- ---------------\n");
fprintf(fp,"\n\n              Item_Name              Item_ID
Inventory_level\n");
fprintf(fp,"   ---------    -------    ---------------\n");
for(i=0;i<a;i++) {

printf("       ");
fprintf(fp,"       ");
printf("%s",info[i].name);
fprintf(fp,"%s",info[i].name);
printf("        %d          %d\n",info[i].id,info[i].i_level);
fprintf(fp,"        %d          %d\n",info[i].id,info[i].i_level);
}
fclose(fp);
getch();
}
```

This page is intentionally left blank