1905648 CC LAB ASSIGNMENT 2

Q-1>Write C program for Min-Min Scheduling Algorithm.

Ans->

```
#include<stdio.h>
#include<limits.h>
int main(){
int nT,nM;//number of tasks , number of machines
printf("Enter number of machines and tasks\n");
scanf("%d %d",&nM,&nT);

int minMin[nM][nT];
int tmp[nM][nT];
int makespan=0;

for(int i=0;i<nM;i++)
{
   printf("For Machine %d \n",i+1);
for(int j=0;j<nT;j++){

printf("Enter Execution Time for Task %d: ",j+1);
scanf("%d",&minMin[i][j]);
tmp[i][j]=minMin[i][j];
}
}
// visualise data
printf("\nOriginal Data\n");

for(int i=0;i<nM;i++){
for(int j=0;j<nT;j++)
printf("%d ",minMin[i][j]);
printf("\n");
}
```

```
//This array will hold the answer

int resultTask[nT];
int resultMachine[nT];
int resultTime[nT];
int ptr = -1; //Indicates if result set is full or not

while(ptr<nT-1){
int time[nT],machine[nT]; //stores minimum time w.r.t
machine of each task
for(int j=0;j<nT;j++){
int minimum = INT_MAX;
int pos=-1;
for(int i=0;i<nM;i++){
if(minMin[i][j]<minimum){
minimum=minMin[i][j];
pos=i;
}
}
time[j]=minimum;
machine[j]=pos;
}

// Now we find task with minimum time

int minimum=INT_MAX;
int pos=-1;

for(int j=0;j<nT;j++){
if(time[j]<minimum){
minimum=time[j];
pos=j;
}
}
```

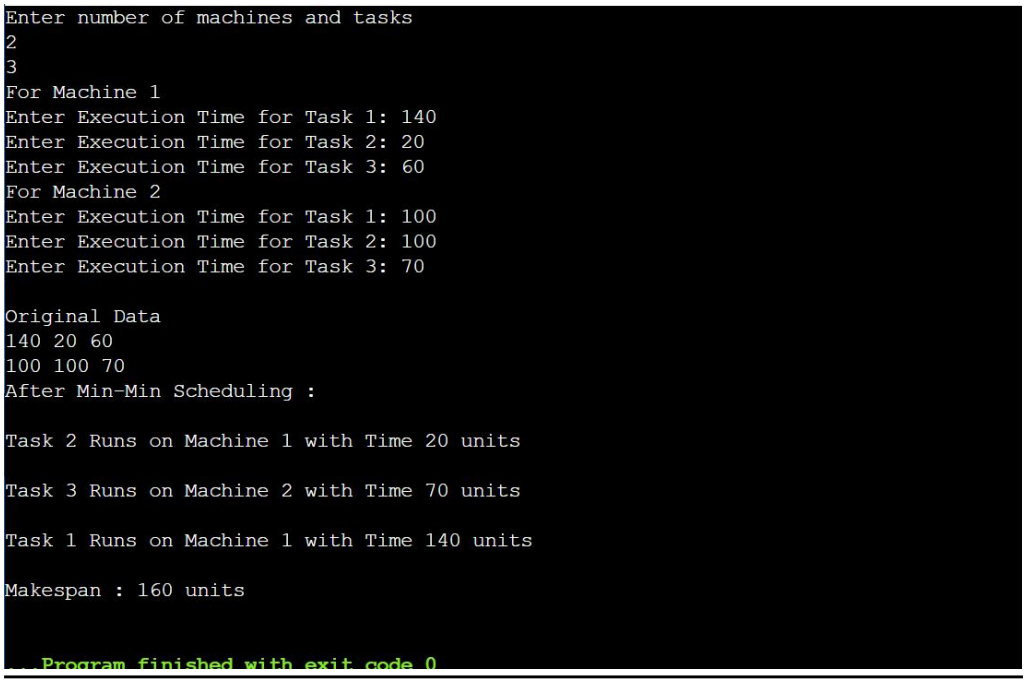
```
resultTask[++ptr]=pos;
resultMachine[ptr]=machine[pos];
resultTime[ptr]=tmp[machine[pos]][pos];
if(minimum>makespan)
makespan=minimum;
// resetting states

for(int i=0;i<nM;i++){
for(int j=0;j<nT;j++){
if(j==resultTask[ptr])
minMin[i][j]=INT_MAX;
else if(i==resultMachine[ptr] && minMin[i][j]!=INT_MAX)
minMin[i][j]+=minimum;
else
continue;
}
}

}
printf("After Min-Min Scheduling :\n");
for(int i=0;i<nT;i++){
printf("\nTask %d Runs on Machine %d with Time %d
units\n",resultTask[i]+1,resultMachine[i]+1,resultTime[i]);
}

printf("\nMakespan : %d units\n",makespan);
return 0;
}
```

OUTPUT



Q-2>Write C program for Max-Min Scheduling Algorithm.

Ans->

```
#include<stdio.h>
#include<limits.h>
int main(){
int nT,nM;//number of tasks , number of machines
printf("Enter number of machines and tasks\n");
scanf("%d %d",&nM,&nT);

int minMin[nM][nT];
int tmp[nM][nT];
int makespan=0;

for(int i=0;i<nM;i++)
{
   printf("For Machine %d \n",i+1);
for(int j=0;j<nT;j++){
```

```
printf("Enter Execution Time for Task %d: ",j+1);
scanf("%d",&minMin[i][j]);
tmp[i][j]=minMin[i][j];
}
}
// visualise data
printf("\nOriginal Data\n");

for(int i=0;i<nM;i++){
for(int j=0;j<nT;j++)
printf("%d ",minMin[i][j]);
printf("\n");
}

//This array will hold the answer

int resultTask[nT];
int resultMachine[nT];
int resultTime[nT];
int ptr = -1; //Indicates if result set is full or not

while(ptr<nT-1){
int time[nT],machine[nT]; //stores minimum time w.r.t
machine of each task
for(int j=0;j<nT;j++){
int minimum = INT_MAX;
int pos=-1;
for(int i=0;i<nM;i++){
if(minMin[i][j]<minimum){
minimum=minMin[i][j];
pos=i;
}
}
time[j]=minimum;
machine[j]=pos;
```

```
}
```

```
// Now we find task with minimum time

int minimum=0;
int pos=-1;

for(int j=0;j<nT;j++){
if(time[j] > minimum){
minimum=time[j];
pos=j;
}
}

resultTask[++ptr]=pos;
resultMachine[ptr]=machine[pos];
resultTime[ptr]=tmp[machine[pos]][pos];
if(minimum>makespan)
makespan=minimum;
// resetting states

for(int i=0;i<nM;i++){
for(int j=0;j<nT;j++){
if(j==resultTask[ptr])
minMin[i][j]=INT_MIN;
else if(i==resultMachine[ptr] && minMin[i][j]!=INT_MIN)
minMin[i][j]+=minimum;
else
```

```
continue;
}
}

}
printf("After Max-Min Scheduling :\n");
for(int i=0;i<nT;i++){
printf("\nTask %d Runs on Machine %d with Time %d
units\n",resultTask[i]+1,resultMachine[i]+1,resultTime[i]);
}

printf("\nMakespan : %d units\n",makespan);
return 0;
}
```

OUTPUT

```
Enter number of machines and tasks
2
3
For Machine 1
Enter Execution Time for Task 1: 140
Enter Execution Time for Task 2: 20
Enter Execution Time for Task 3: 60
For Machine 2
Enter Execution Time for Task 1: 100
Enter Execution Time for Task 2: 100
Enter Execution Time for Task 3: 70

Original Data
140 20 60
100 100 70
After Max-Min Scheduling :

Task 1 Runs on Machine 2 with Time 100 units

Task 3 Runs on Machine 1 with Time 60 units

Task 2 Runs on Machine 1 with Time 20 units

Makespan : 100 units
```