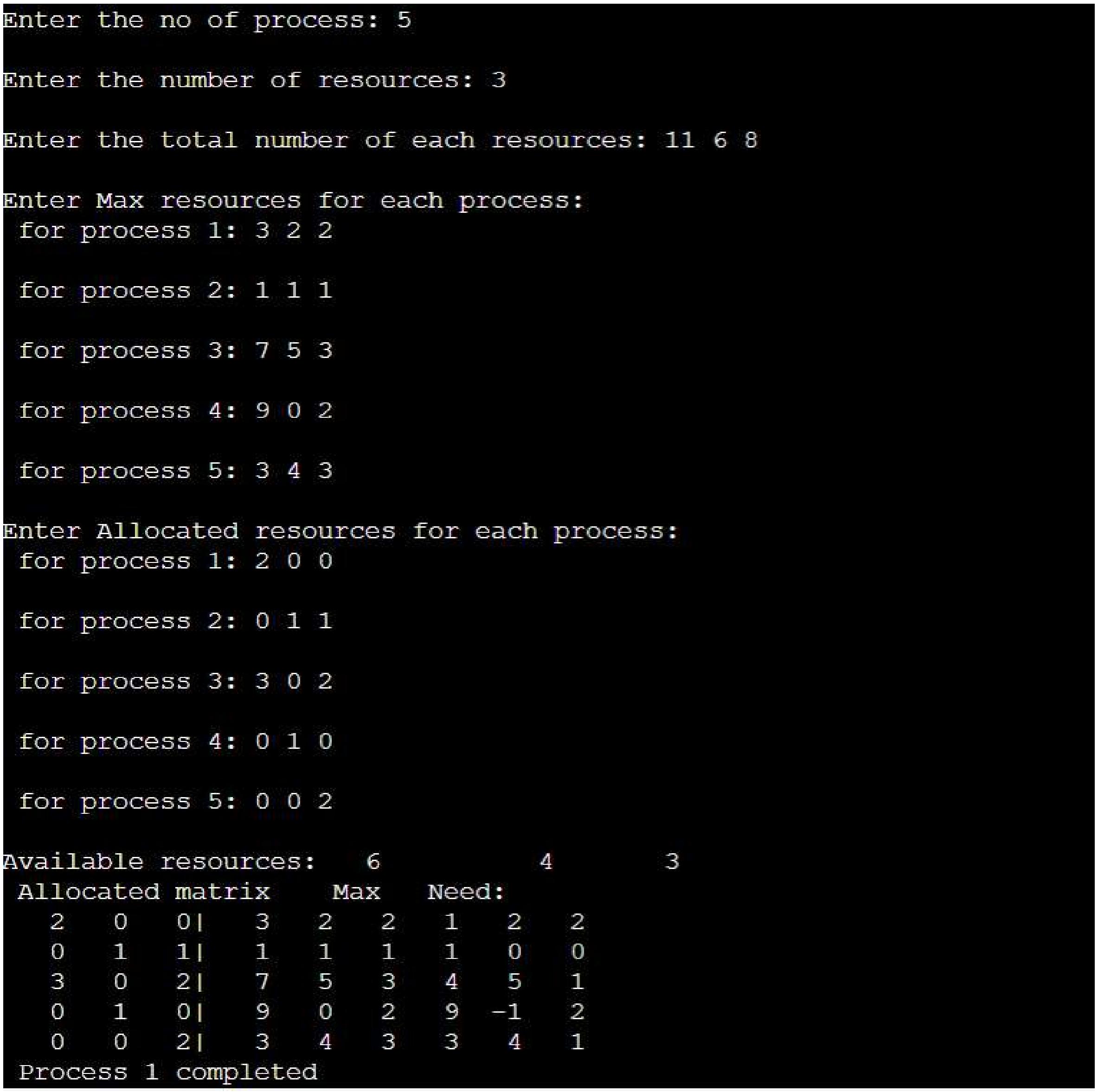
Program:

Aim :Write a C program to simulate the Banker’s Algorithm for Deadlock Avoidance.

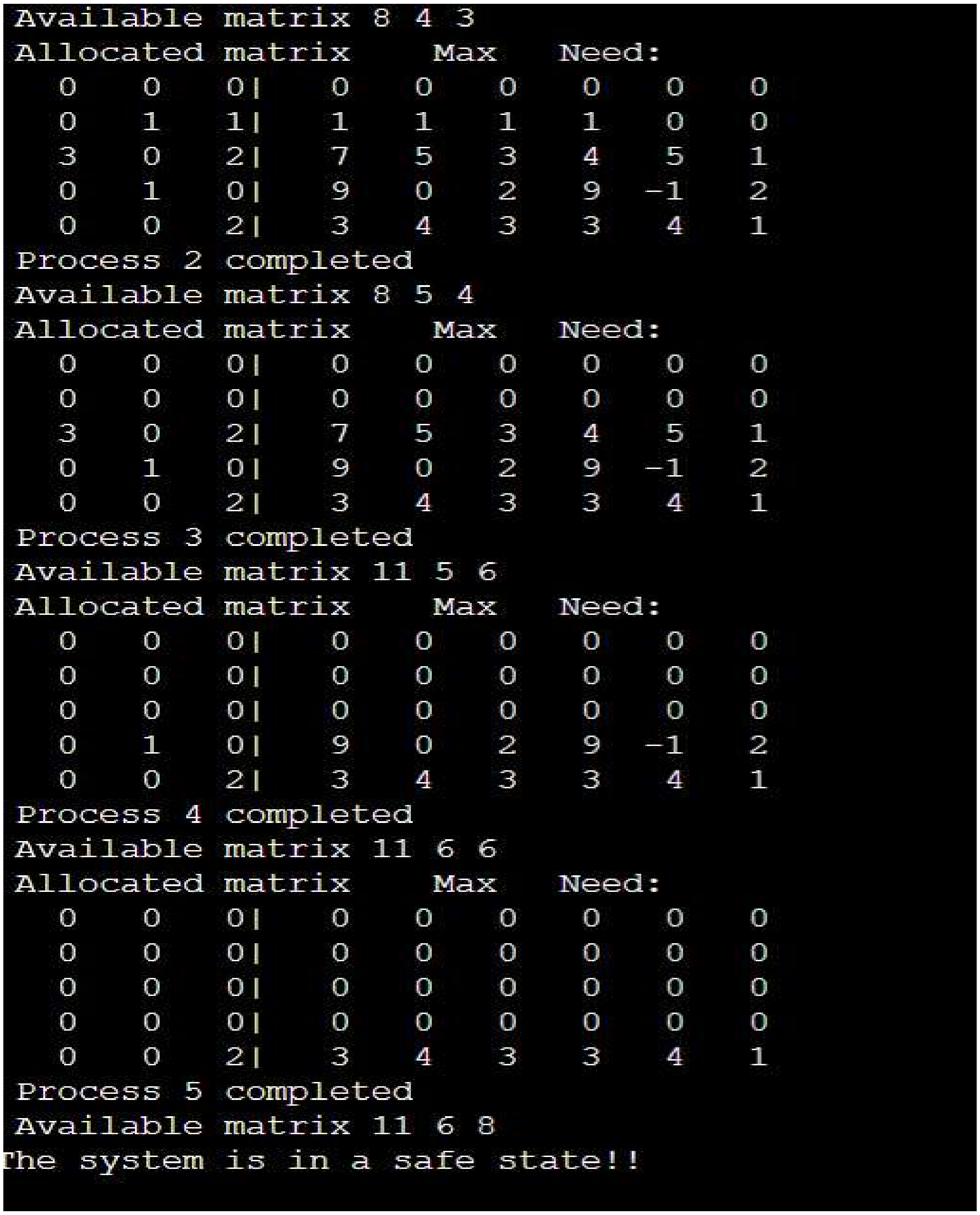
```
#include<stdio.h>
int main()
{
int allocated[15][15],max[15][15],need[15][15],avail[15],tres[15],work[15],flag[15];
int pno,rno,i,j,prc,total,count=0;
printf("\nEnter the no of process: ");
scanf("%d",&pno);
printf("\nEnter the number of resources: ");
scanf("%d",&rno);
for(i=1;i<=pno;i++)
{
flag[i]=0;
}
printf("\nEnter the total number of each resources: ");
for(i=1;i<=rno;i++)
{
scanf("%d",&tres[i]);
}
printf("\nEnter Max resources for each process: ");
for(i=1;i<=pno;i++)
{
printf("\n for process %d: ", i);
for(j=1;j<=rno;j++)
{
scanf("%d",&max[i][j]);
}
}
printf("\nEnter Allocated resources for each process: ");
for(i=1;i<=pno;i++)
{
printf("\n for process %d: ", i);
for(j=1;j<=rno;j++)
{
scanf("%d",&allocated[i][j]);
}
}
printf("\nAvailable resources: ");
for(j=1;j<=rno;j++)
{
avail[j]=0;
total=0;
for(i=1;i<=pno;i++)
{
total+=allocated[i][j];
}
avail[j] = tres[j]-total;
work[j] = avail[j];
printf("  %d\t",work[j]);
}
do
```

```
{
for(i=1;i<=pno;i++)
{
for(j=1;j<=rno;j++)
{
need[i][j] = max[i][j]-allocated[i][j];
}
}
printf("\n Allocated matrix    Max   Need: ");
for(i=1;i<=pno;i++)
{
printf("\n");
for(j=1;j<=rno;j++)
{
printf("%4d",allocated[i][j]);
}
printf("|");
for(j=1;j<=rno;j++)
{
printf("%4d",max[i][j]);
}
for(j=1;j<=rno;j++)
{
printf("%4d",need[i][j]);
}
}
prc=0;
for(i=1;i<=pno;i++)
{
if(flag[i]==0)
{
prc=i;
for(j=1;j<=rno;j++)
{
if(work[j]<need[i][j])
{
prc=0;
break;
}
}
}
if(prc!=0)
break;
}
if(prc!=0)
{
printf("\n Process %d completed",i);
count++;
printf("\n Available matrix ");
for(j=1;j<=rno;j++)
{
work[j]+=allocated[prc][j];
allocated[prc][j] = 0;
max[prc][j] = 0;
flag[prc] = 1;
printf ("%d ", work[j]);
}
}
}while(count != pno&&prc != 0);
if (count == pno)
printf ("\nThe system is in a safe state!!");
else
printf ("\nThe system is in an unsafe state!!");
return 0;
}
```

Output:



```
Enter the no of process: 5

Enter the number of resources: 3

Enter the total number of each resources: 11 6 8

Enter Max resources for each process:
 for process 1: 3 2 2

 for process 2: 1 1 1

 for process 3: 7 5 3

 for process 4: 9 0 2

 for process 5: 3 4 3

Enter Allocated resources for each process:
 for process 1: 2 0 0

 for process 2: 0 1 1

 for process 3: 3 0 2

 for process 4: 0 1 0

 for process 5: 0 0 2

Available resources:   6          4       3
 Allocated matrix    Max   Need:
   2   0   0|   3   2   2   1   2   2
   0   1   1|   1   1   1   1   0   0
   3   0   2|   7   5   3   4   5   1
   0   1   0|   9   0   2   9  -1   2
   0   0   2|   3   4   3   3   4   1
 Process 1 completed
```



```
Available matrix 8 4 3
Allocated matrix    Max   Need:
  0   0   0|   0   0   0   0   0   0
  0   1   1|   1   1   1   1   0   0
  3   0   2|   7   5   3   4   5   1
  0   1   0|   9   0   2   9  -1   2
  0   0   2|   3   4   3   3   4   1
Process 2 completed
Available matrix 8 5 4
Allocated matrix    Max   Need:
  0   0   0|   0   0   0   0   0   0
  0   0   0|   0   0   0   0   0   0
  3   0   2|   7   5   3   4   5   1
  0   1   0|   9   0   2   9  -1   2
  0   0   2|   3   4   3   3   4   1
Process 3 completed
Available matrix 11 5 6
Allocated matrix    Max   Need:
  0   0   0|   0   0   0   0   0   0
  0   0   0|   0   0   0   0   0   0
  0   0   0|   0   0   0   0   0   0
  0   1   0|   9   0   2   9  -1   2
  0   0   2|   3   4   3   3   4   1
Process 4 completed
Available matrix 11 6 6
Allocated matrix    Max   Need:
  0   0   0|   0   0   0   0   0   0
  0   0   0|   0   0   0   0   0   0
  0   0   0|   0   0   0   0   0   0
  0   0   0|   0   0   0   0   0   0
  0   0   2|   3   4   3   3   4   1
Process 5 completed
Available matrix 11 6 8
The system is in a safe state!!
```