Write a C program to simulate Bankers algorithm for the purpose of Deadlock avoidance.

```
# include <Stdio.h>
# include <conio.h>
# include <string.h>
Void main ()
{
 int alloc [10][10],
 max [10][10];
 int avail [10], work [10], total [10];
 int i, j, k, n, need [10][10];
 int m;
 int count=0, c=0;
 Char finish [10];
 printf ("Enter the no of processes and resources:");
 Scanf ("%d %d", &n, &m);
 for (i=0; i<=n; i++) finish [i]='n';
 printf (" Enter the claim matrix :\n");
 for (i=0 ; i<n; i++)
 for (j=0 ; j<m ; j++)
 Scanf ("%d", &alloc [i][j]);
 printf ("Resource vector :"); for (i=0; i<m; i++)
  Scanf ("%d", & total (i]); for (i=0; i<m; i++)
  avail [i]=0; for(i=0; i<n; i++)

for (j=0; j<m; j++) avail [j] +=alloc [i][j];
for (i=0; i<m; i++) work [i] = avail [i];
for (j=0; j<m; j++) work [i] =total [j]-work[j
for (i=0; i<n; i++)
   for (j=0; j<m; j++) need [i][j]=max[i][j]
                      -alloc [i][j]; 1:

for(i=0; i<n; i++)
{
 c=0;
 for (j=0; j<m; j++)
 if ( (need[i][j] <= work [j]) && (finish [i]=='n')) c++;
 if (c==m)
 {
 printf ("All the resources can be allocated to process %d",
                                                         i+1);
 printf ("\n\nAvailable resources are:");
   for (k=0; k<m; k++)
 {
  work[k]+=alloc [i][k];
  printf ("%4d", work [k]);
 }
 printf ("\n");
 finish[i]='y';
 printf ("\nprocess %d executed ? : %c\n", i+1, finish[i]);
   count ++;
 }
 }
 if (count !=n) goto A;
 else
 printf ("\n System is in safe mode");
 printf ("\n The given state is safe state");
   getch();
 }
```

output :

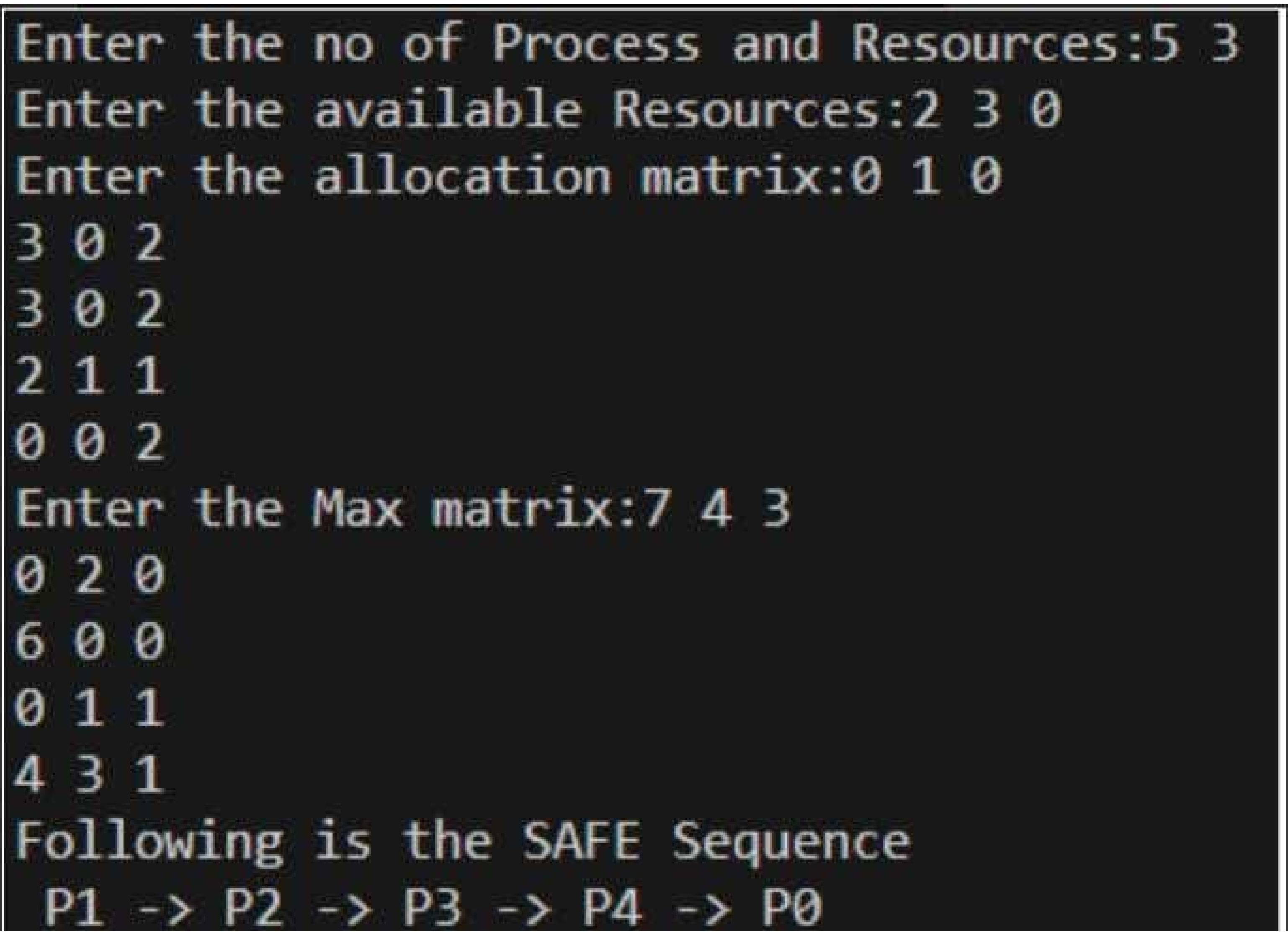
```
Enter the no. of process and resources: 4 3
Enter the claim matrix:
 3  2  2
 6  1  3
 3  1  4
 4  2  2
```

```
Entr the allocation matrix :
  1  0  0
  6  1  2
  2  1  1
  0  0  2
Resoure vector : 9  3  6
All the resources can be allocated to process 2
Available resources are : 6  2  3
process 2 executed ? : y
All the resource can be allocated to process 3
Available resources are : 8  3  4
process 3 executed ? : y
All the resources can be allocated to process 4
Available resources are  8  3  6
process 4 executed ? : y
All the resources can be allocated to process 1
Available resources are  9  3  6
process 1 executed ? : y

System is in safe mode
The given state is safe state.
```

10/10

2/8/23



```
Enter the no of Process and Resources:5 3
Enter the available Resources:2 3 0
Enter the allocation matrix:0 1 0
3 0 2
3 0 2
2 1 1
0 0 2
Enter the Max matrix:7 4 3
0 2 0
6 0 0
0 1 1
4 3 1
Following is the SAFE Sequence
 P1 -> P2 -> P3 -> P4 -> P0
```