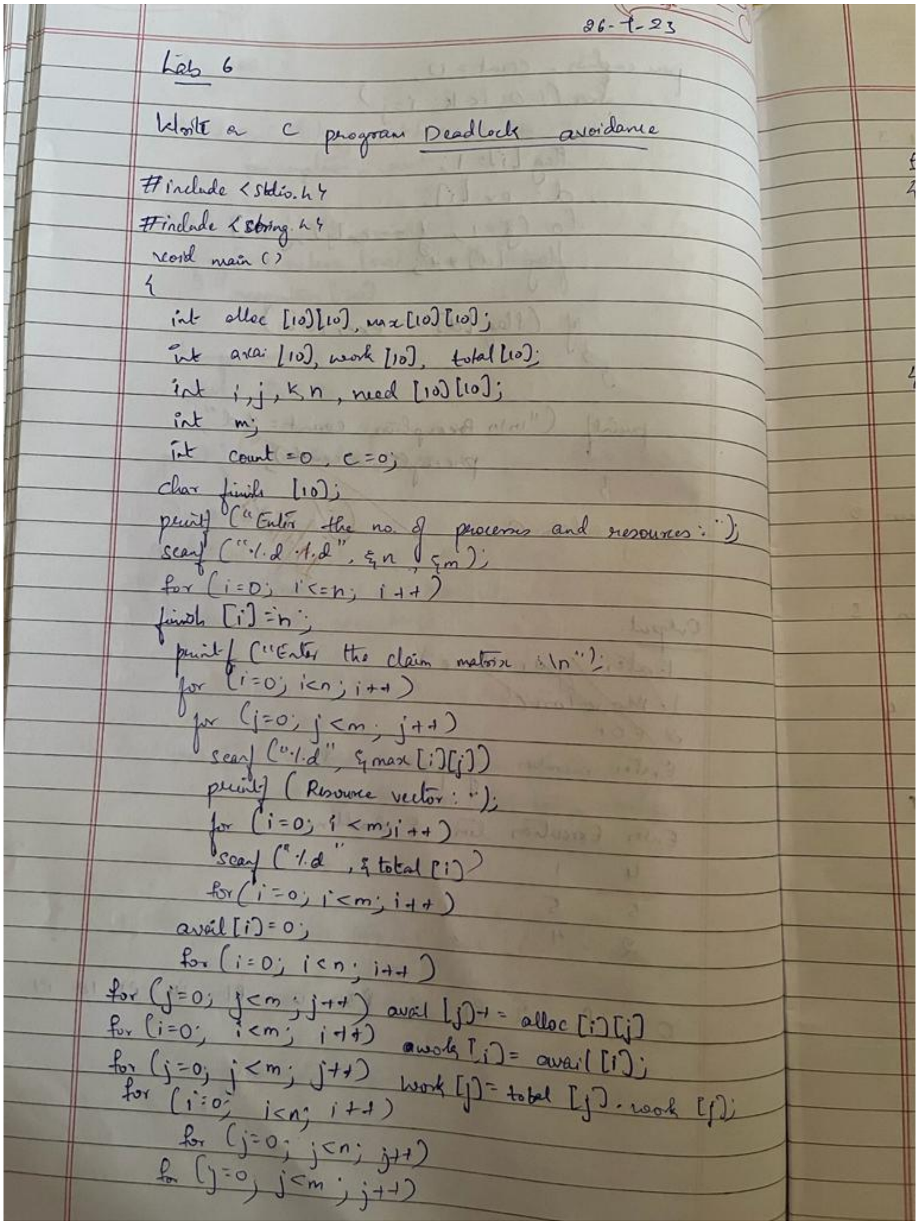
LAB 6

To Simulate bankers algorithm for DeadLock Avoidance (Banker‘s Algorithm)



26-7-23

Lab 6

Write a C program Deadlock avoidance

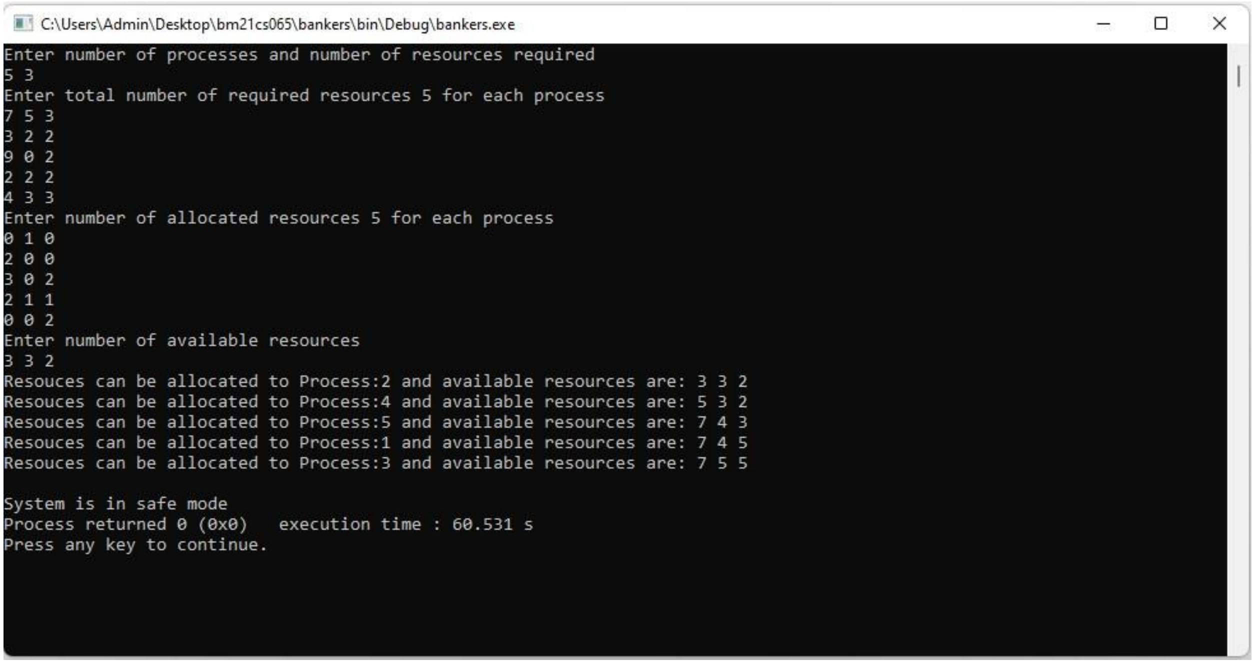
```
#include <stdio.h>
#include <string.h>
void main ()
{
   int alloc [10][10], max[10][10];
   int avai [10], work [10], total [10];
   int i,j,k,n , need [10][10];
   int m;
   int count =0 , c=0;
  char finish [10];
  printf ("Enter the no. of processes and resources:");
  scanf ("%d %d", &n , &m);
  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", &max [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 [j]= total [j]-work [j];
 for (i=0; i<n; i++)
    for (j=0; j<n; j++)
  for (j=0, j<m ; j++)
```

```
    need [i][j] = max [i] [j]
     alloc [i] [i] = A >
for (i=0; i<n; i++)
{
    c = 0;
    for (i=0; j<m; j+)
   if [(need[i][j] <= work [j]) && [finish [i] = n')]
     c++;
  if (c==m)
{
printf ("All the resource can be allocated to process
        %d", i+1);
printf ("\n\n Available resource are: "):
    for (k=0) k<m; k++)
    {
      work [K] += alloc [i][K];
      printf ("%w", work [K]);
    }

    printf ("\n");
    finish [i] = 'y';
  printf ("\n process %d executed? %6\n
         i+1, finish [i]);
  count ++;
}
if (count != n) goto n;
else
printf ("\n system is in safe mode");
printf ("\n The given state is safe state);
getch ();
}
```

Output:

```
Enter the no of process and resource : 4 3
Enter the claim matrix:
        3   2   2
        6   1   3
        3   1   4
        4   2   2
Enter the allocation matrix:
          1  0  0
          6  1  2
          2  1  1
          0  0  2
Resource vetor : 9 3 6
All the resources can be allocated to process 9
Available resources are : 6 2 3
Process 2 executed ? : y
All the resources can be allocated to process 3
Available resources are : 3 4
Process 3 executed ? : y
All the resources can be allocated to process 4
Available resource are 8 3 6
process 4 executed ? : y
All the resources can be allocated to process 1
Available resources are 9 8 6
process 1 executed ? : y

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

OUTPUT:


C:\Users\Admin\Desktop\bm21cs065\bankers\bin\Debug\bankers.exe

```
Enter number of processes and number of resources required
5 3
Enter total number of required resources 5 for each process
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter number of allocated resources 5 for each process
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter number of available resources
3 3 2
Resouces can be allocated to Process:2 and available resources are: 3 3 2
Resouces can be allocated to Process:4 and available resources are: 5 3 2
Resouces can be allocated to Process:5 and available resources are: 7 4 3
Resouces can be allocated to Process:1 and available resources are: 7 4 5
Resouces can be allocated to Process:3 and available resources are: 7 5 5

System is in safe mode
Process returned 0 (0x0)   execution time : 60.531 s
Press any key to continue.
```