

## BANKER'S ALGORITHM

AIM:

TO WRITE A PROGRAM FOR BANKER'S ALGORITHM USING C LANGUAGE.

**Banker's Algorithm** is used majorly in the banking system to avoid deadlock. It helps you to identify whether a loan will be given or not.

This algorithm is used to test for safely simulating the allocation for determining the maximum amount available for all resources.

It also checks for all the possible activities before determining whether allocation should be continued or not.

ALGORITHM:

Write a C Program to implement Bankers algorithm for deadlock avoidance?

For example:

Test	Input	Result
T1	3	Max matrix:      Allocation matrix:
	3	3 3 3                      3 2 1
	3	3 3 3                      2 2 2
	3	3 3 3                      3 3 3
	3	
	3	Process 1 runs to completion!
	3	Max matrix:      Allocation matrix:
	3	0 0 0                      0 0 0
	3	3 3 3                      2 2 2
	3	3 3 3                      3 3 3
	3	
	3	Process 2 runs to completion!
	2	Max matrix:      Allocation matrix:
	1	0 0 0                      0 0 0
	2	0 0 0                      0 0 0
	2	3 3 3                      3 3 3
	2	
	3	Process 3 runs to completion!
	3	The system is in a safe state!!
	3	Safe Sequence : < 0 1 2 >
	4	
	1	
	2	

PROGRAM:

```

1  #include<stdio.h>
2  int main()
3  {
4      int n,m;
5      scanf("%d%d",&n,&m);
6      int max[3][3],all[3][3];
7      for(int i=0;i<3;i++)
8          for(int j=0;j<3;j++)
9              scanf("%d",&max[i][j]);
10     for(int i=0;i<3;i++)
11         for(int j=0;j<3;j++)
12             scanf("%d",&all[i][j]);
13     int ave[3];
14     for(int i=0;i<3;i++)
15         scanf("%d",&ave[i]);
16     int v[3]={-1,-1,-1};
17     int f=1;
18     int p=0;
19     while(f)
20     {
21         f=0;
22         for(int i=0;i<3;i++)
23

```

Answer: (penalty regime: 0 %)

```

19     while(f)
20     {
21         f=0;
22         for(int i=0;i<3;i++)
23         {
24             if(v[i]==-1)
25             {
26                 f=1;
27                 int s=0;
28                 for(int j=0;j<3;j++)
29                 {
30                     if(max[i][j]<=all[i][j]+ave[j])
31                         s+=1;
32                 }
33                 if(s==3)
34                 {
35                     v[i]=p;
36                     p+=1;
37                     printf("Max matrix:\tAllocation matrix:\n");
38                     if(p<=n)
39                         for(int j=0;j<3;j++)
40                         {
41

```

Answer: (penalty regime: 0 %)

```
39     for(int j=0;j<3;j++)
40     {
41         for(int k=0;k<3;k++)
42         {
43             printf("%d ",max[j][k]);
44         }
45         printf("\t\t");
46         for(int k=0;k<3;k++){
47             printf("%d ",all[j][k]);
48         }
49         printf("\n");
50     }
51     printf("\nProcess %d runs to completion!\n",p);
52     if(p>0 && p<n)
53         printf(" ");
54     for(int j=0;j<3;j++)
55     {
56         max[i][j]=0;
57         ave[i]=all[i][j] + ave[i];
58         all[i][j]=0;
59     }
60 }
61 }
```



Answer: (penalty regime: 0 %)

```
45     printf("\t\t");
46     for(int k=0;k<3;k++){
47         printf("%d ",all[j][k]);
48     }
49     printf("\n");
50 }
51 printf("\nProcess %d runs to completion!\n",p);
52 if(p>0 && p<n)
53     printf(" ");
54 for(int j=0;j<3;j++)
55 {
56     max[i][j]=0;
57     ave[i]=all[i][j] + ave[i];
58     all[i][j]=0;
59 }
60 }
61 }
62 }
63 }
64 printf("The system is in a safe state!!\n");
65 printf("Safe Sequence : < %d %d %d >",v[0],v[1],v[2]);
66 }
```

OUTPUT:

	Test	Input	Expected	Got
✓	T1	3	Max matrix:\tAllocation matrix:	Max matrix:\tAllocation matrix:
		3	3 3 3 \t\t3 2 1	3 3 3 \t\t3 2 1
		3	3 3 3 \t\t2 2 2	3 3 3 \t\t2 2 2
		3	3 3 3 \t\t3 3 3	3 3 3 \t\t3 3 3
		3		
		3	Process 1 runs to completion!	Process 1 runs to completion!
		3	Max matrix:\tAllocation matrix:	Max matrix:\tAllocation matrix:
		3	0 0 0 \t\t0 0 0	0 0 0 \t\t0 0 0
		3	3 3 3 \t\t2 2 2	3 3 3 \t\t2 2 2
		3	3 3 3 \t\t3 3 3	3 3 3 \t\t3 3 3
		3		
		3	Process 2 runs to completion!	Process 2 runs to completion!
		2	Max matrix:\tAllocation matrix:	Max matrix:\tAllocation matrix:
		1	0 0 0 \t\t0 0 0	0 0 0 \t\t0 0 0
		2	0 0 0 \t\t0 0 0	0 0 0 \t\t0 0 0
		2	3 3 3 \t\t3 3 3	3 3 3 \t\t3 3 3
		2		

RESULT:

GIVEN PROGRAM FOR BANKER'S ALGORITHM USING C LANGUAGE WAS SUCCESSFULLY EXECUTED.