

Write a C-program to simulate Banker's 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", &max[i][j]);
    printf("Enter the allocation 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[j] = 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]; A;
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\n Available resources are: ");
        for (k=0; k<m; k++)
        {
            work[k] += alloc[i][k];
            printf("%d ", work[k]);
        }
        printf("\n");
        finish[i] = 'y';
        printf("\n Process %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 processes and resources: 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 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 resources 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

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

N
2/5/23

```
2 2
Enter the allocation matrix:
0 0
5 1 2
2 1 1
0 0 2
Resource 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 resources 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_
```