

Bankers Theorem Yash Singhal 2400290120285

```
bankers_Theorem.c
unn.term

Save TimeTravel Assistant Server Code File Edit Format View Help

bankers_Theorem.c
1 #include <stdio.h>
2
3 int main() {
4     int n, m;
5     printf("Enter the no of processes: ");
6     scanf("%d", &n);
7     printf("Enter the no of resources: ");
8     scanf("%d", &m);
9
10    int alloc[n][m], max[n][m], avail[m];
11
12    printf("Enter the allocation matrix:\n");
13    for (int i = 0; i < n; i++) {
14        for (int j = 0; j < m; j++) {
15            scanf("%d", &alloc[i][j]);
16        }
17    }
18
19    printf("Enter the maximum matrix:\n");
20    for (int i = 0; i < n; i++) {
21        for (int j = 0; j < m; j++) {
22            scanf("%d", &max[i][j]);
23        }
24    }
25
26    printf("Enter the available resources:\n");
27    for (int i = 0; i < m; i++) {
28        scanf("%d", &avail[i]);
29    }
30
31    int f[n], ans[n], ind = 0;
32    for (int k = 0; k < n; k++) f[k] = 0;
33
34    int need[n][m];
35    for (int i = 0; i < n; i++) {
36        for (int j = 0; j < m; j++) {
37            need[i][j] = max[i][j] - alloc[i][j];
38        }
39    }
40
41    for (int k = 0; k < n; k++) {
42        for (int i = 0; i < n; i++) {
43            if (f[i] == 0) {
44                int flag = 0;
45                for (int j = 0; j < m; j++) {
46                    if (need[i][j] > avail[j]) {
47                        flag = 1;
48                        break;
49                    }
50                }
51                if (flag == 0) {
52                    ans[ind++] = i;
53                    for (int y = 0; y < m; y++) {
54                        avail[y] += alloc[i][y];
55                    }
56                    f[i] = 1;
57                }
58            }
59        }
60    }
61
62    int safe = 1;
63    for (int i = 0; i < n; i++) {
64        if (f[i] == 0) {
65            safe = 0;
66            break;
67        }
68    }
69
70    if (safe) {
71        printf("The SAFE Sequence is as follows:\n");
72        for (int i = 0; i < n - 1; i++)
73            printf(" P%d ->", ans[i]);
74        printf(" P%d\n", ans[n - 1]);
75    } else {
76        printf("The system is in Deadlock.!\n");
77    }
78
79    return 0;
80 }
81
```

```
bankers_Therorem.c × unn.term × +
Assistant Server ~ Terminal File Edit View Go Help
~$ ./banks.txt
Enter the no of processes: 5
Enter the no of resources: 3
Enter the allocation matrix:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the maximum matrix:
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter the available resources:
3 3 2
The SAFE Sequence is as follows:
P1 -> P3 -> P4 -> P0 -> P2
~$ ./banks.txt
Enter the no of processes: 3
Enter the no of resources: 1
Enter the allocation matrix:
2 3 1
Enter the maximum matrix:
4 5 3
Enter the available resources:
1
The system is in Deadlock.!
```

```
~$ ./banks.txt
Enter the no of processes: 3
Enter the no of resources: 1
Enter the allocation matrix:
2 3 1
Enter the maximum matrix:
4 5 3
Enter the available resources:
1
The system is in Deadlock.!
```