Write a C program to simulate: (Any one)
b) Deadlock Detection

```
#include <stdio.h>
#include <stdbool.h>

#define MAX 10

int main() {
    int n, m, i, j;
    int Allocation[MAX][MAX], Request[MAX][MAX], Available[MAX], Work[MAX];
    bool Finish[MAX];
    int safeSequence[MAX];
    int safeIndex = 0;

    printf("Processes: "); scanf("%d", &n);
    printf("Resources: "); scanf("%d", &m);

    printf("Allocation (%dx%d):\n", n, m);
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            scanf("%d", &Allocation[i][j]);

    printf("Request (%dx%d):\n", n, m);
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            scanf("%d", &Request[i][j]);

    printf("Available (%d):\n", m);
    for (j = 0; j < m; j++)
        scanf("%d", &Available[j]);

    for (j = 0; j < m; j++) Work[j] = Available[j];
    for (i = 0; i < n; i++) Finish[i] = false;

    bool progress = true;
    while (progress) {
        progress = false;
        for (i = 0; i < n; i++) {
```

```
            if (!Finish[i]) {
                bool canRun = true;
                for (j = 0; j < m; j++)
                    if (Request[i][j] > Work[j]) {
                        canRun = false;
                        break;
                    }

                if (canRun) {
                    for (j = 0; j < m; j++)
                        Work[j] += Allocation[i][j];
                    Finish[i] = true;
                    safeSequence[safeIndex++] = i;
                    progress = true;
                }
            }
        }
    }
    bool deadlock = false;
    for (i = 0; i < n; i++) {
        if (!Finish[i]) {
            deadlock = true;
            printf("Deadlock detected! Process %d is deadlocked.\n", i);
        }
    }

    if (!deadlock) {
        printf("No deadlock detected.\nSafe Sequence: ");
        for (i = 0; i < n; i++) {
            printf("P%d ", safeSequence[i]);
        }
        printf("\n");
    }

    return 0;
}
```

Output:

With deadlock

```
Processes: 5
Resources: 3
Allocation (5x3):
0 1 0
2 0 0
3 0 3
2 1 1
0 0 2
Request (5x3):
0 0 0
2 0 2
0 0 1
1 0 0
0 0 2
Available (3):
0 0 0
Deadlock detected! Process 1 is deadlocked.
Deadlock detected! Process 2 is deadlocked.
Deadlock detected! Process 3 is deadlocked.
Deadlock detected! Process 4 is deadlocked.
```

No deadlock

```
Processes: 5
Resources: 3
Allocation (5x3):
0 1 0
2 0 0
3 0 3
2 1 1
0 0 2
Request (5x3):
0 0 0
2 0 2
0 0 0
1 0 0
0 0 2
Available (3):
0 0 0
No deadlock detected.
Safe Sequence: P0 P2 P3 P4 P1
```