

DEADLOCK:

```
#include <stdio.h>
#define MAX 10 // Max processes and resources
int isSafeState(int available[], int max[][MAX], int allot[][MAX], int P, int R) {
    int need[P][R], work[R], finish[P]
    for (int i = 0; i < P; i++) finish[i] = 0; // Initialize finish to 0
    for (int i = 0; i < R; i++) work[i] = available[i]; // Initialize work with available resource;
    for (int i = 0; i < P; i++)
        for (int j = 0; j < R; j++) need[i][j] = max[i][j] - allot[i][j]; // Need = Max - Allocated
    int count = 0;
    while (count < P) {
        int found = 0;
        for (int p = 0; p < P; p++) {
            if (!finish[p]) {
                int canFinish = 1;
                for (int j = 0; j < R; j++) if (need[p][j] > work[j]) { canFinish = 0; break; }
                if (canFinish) {
                    for (int j = 0; j < R; j++) work[j] += allot[p][j]; // Add allocated resources to work
                    finish[p] = 1; count++; found = 1;
                }
            }
        }
        if (!found) return 0; // Unsafe state
    }
    return 1; // Safe state
}

int main() {
    int P, R;
    printf("Enter the number of processes: ");
    scanf("%d", &P);
    printf("Enter the number of resources: ");
    scanf("%d", &R);

    int available[R], max[P][R], allot[P][R];

    printf("Enter available resources: ");
    for (int i = 0; i < R; i++) scanf("%d", &available[i]);

    printf("Enter max resources for each process:\n");
    for (int i = 0; i < P; i++) for (int j = 0; j < R; j++) scanf("%d", &max[i][j]);

    printf("Enter allocated resources for each process:\n");
    for (int i = 0; i < P; i++) for (int j = 0; j < R; j++) scanf("%d", &allot[i][j]);
}
```

```
if (isSafeState(available, max, allot, P, R)) printf("The system is in a safe state.\n");  
else printf("The system is in an unsafe state.\n");  
  
return 0;  
}
```