Name : Ritesh Rathod
Roll No. : TYCOD209

```
//Banker's Algorithm
#include <stdio.h>
#include <stdbool.h>

void findSafeSequence(int n, int m, int allocated[n][m], int maxRequire[n][m], int instances[m],
char process[n]) {
    int need[n][m];
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            need[i][j] = maxRequire[i][j] - allocated[i][j];
        }
    }

    int available[m];
    for(int i = 0; i < m; i++) {
        int sum = 0;
        for(int j = 0; j < n; j++) {
            sum += allocated[j][i];
        }
        available[i] = instances[i] - sum;
    }

    bool finish[n];
    for(int i = 0; i < n; i++) {
        finish[i] = false;
    }

    int safeSequence[n];
    int work[m];
    for(int i = 0; i < m; i++) {
        work[i] = available[i];
    }

    int count = 0;
    while(count < n) {
        bool found = false;
        for(int i = 0; i < n; i++) {
            if(finish[i]==false) {
                bool canProcess = true;
                for(int j = 0; j < m; j++) {
                    if(need[i][j] > work[j]) {
                        canProcess = false;
                        break;
                    }
                }
```

```
            if(canProcess) {
                for(int j = 0; j < m; j++) {
                    work[j] += allocated[i][j];
                }
                safeSequence[count++] = i;
                finish[i] = true;
                found = true;
            }
        }
    }
    if(found == false) {
        printf("Unsafe state! System is not in a safe state.\n");
        return;
    }
}

printf("Safe sequence: ");
for(int i = 0; i < n; i++) {
    printf("%c ", process[safeSequence[i]]);
}
printf("\n");
}



int main(void) {

    int n,m;
    printf("Enter the number of processes: ");
    scanf("%d",&n);

    char process[n];
    printf("Enter the process names: ");
    for(int i=0;i<n;i++){
        scanf(" %c",&process[i]);
    }

    printf("Enter the number of resources: ");
    scanf("%d",&m);

    int instances[m];
    printf("Enter the total instances of each resource:\n");
    for(int i=0;i<m;i++){
        printf("Resource %d: ",i+1);
        scanf("%d",&instances[i]);
    }

    int maxRequire[n][m];
```

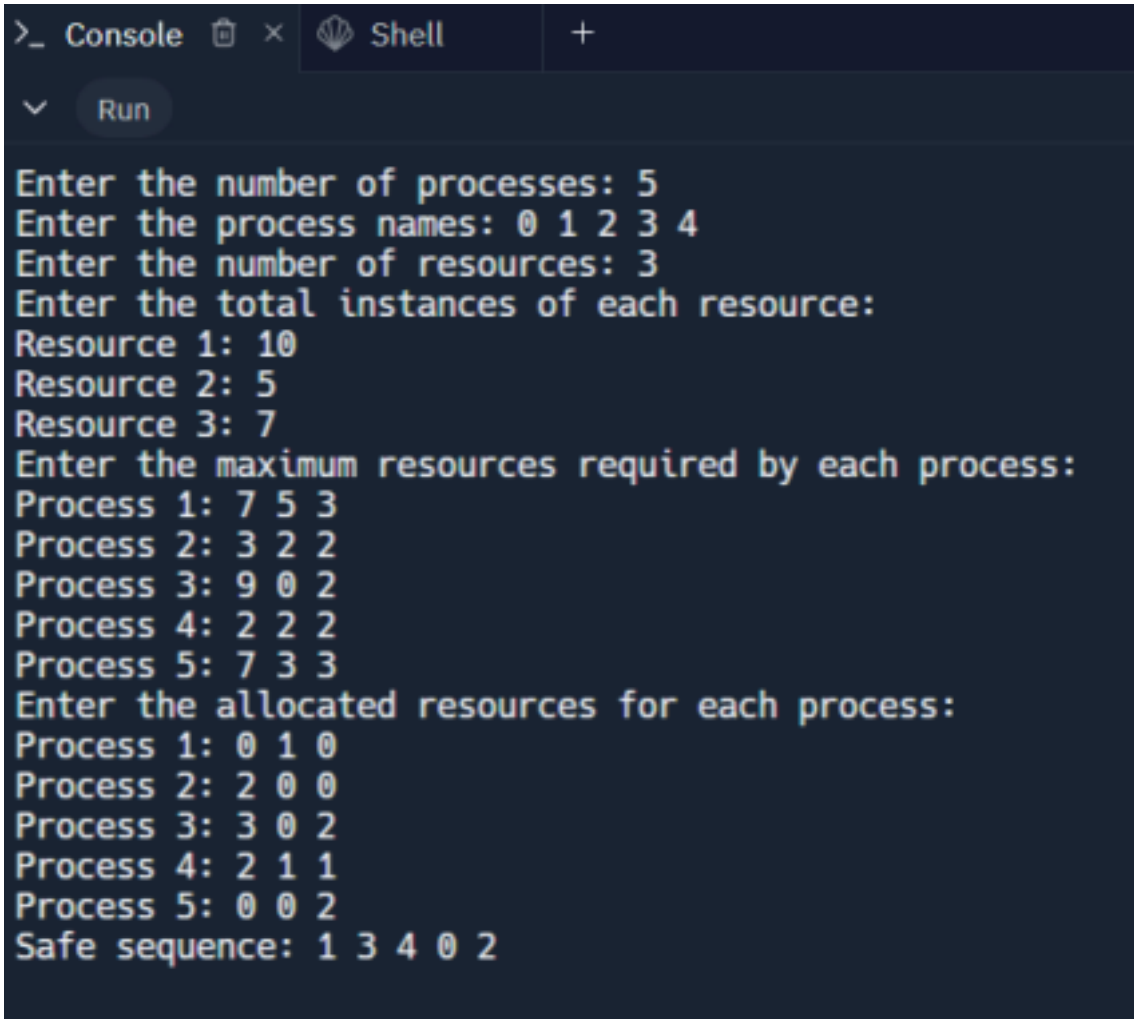
```
    printf("Enter the maximum resources required by each process:\n");
    for(int i=0;i<n;i++){
        printf("Process %d: ",i+1);
        for(int j=0;j<m;j++){
            scanf("%d",&maxRequire[i][j]);
        }
    }

    int allocated[n][m];
    printf("Enter the allocated resources for each process:\n");
    for(int i=0;i<n;i++){
        printf("Process %d: ",i+1);
        for(int j=0;j<m;j++){
            scanf("%d",&allocated[i][j]);
        }
    }



    findSafeSequence(n, m, allocated, maxRequire, instances, process);

    return 0;
}
```


Console
Shell
Run
Enter the number of processes: 5
Enter the process names: 0 1 2 3 4
Enter the number of resources: 3
Enter the total instances of each resource:
Resource 1: 10
Resource 2: 5
Resource 3: 7
Enter the maximum resources required by each process:
Process 1: 7 5 3
Process 2: 3 2 2
Process 3: 9 0 2
Process 4: 2 2 2
Process 5: 7 3 3
Enter the allocated resources for each process:
Process 1: 0 1 0
Process 2: 2 0 0
Process 3: 3 0 2
Process 4: 2 1 1
Process 5: 0 0 2
Safe sequence: 1 3 4 0 2