Q1 C code to implement Bankers algorithm

```
#include <stdio.h>
#include <stdbool.h>
# define MAX_PROCESSES 10
# define MAX_RESOURCES 10

void calculatedNeed(int need[MAX_PROCESSES]
        [MAX_RESOURCES], int max [MAXPROCESSES]
        [MAX_RESOURCES], int allot[MAX_PROCESSES]
        [MAX_RESOURCES], int np int nr) {
    for (int i=0; i<np; i++) {
        for (int j=0; j<nr; j++) {
            need[i][j]=max[i][j]- allot[i][j];
        }
    }
}

bool isSafe(int processes[], int avail[], int
    max[][MAX_RESOURCES], int allot[][MAX_
    RESOURCES], int np, int nr);
    bool finish [MAX_PROCESSES]= {0};
    int safeSeq [MAX_PROCESSES];
    int work[MAX_RESOURCES];
    for (int i=0; i<nr; i++)
        work[i]= avail [i];
    int count=0;
    while (count<np) {
        bool found= false;
        for (int P=0; P<np; P++) {
            if (finish [P]==0) {
                int j;
                for(j=0; j<nr; j++)
                    if (need [P][j]> work[j])
                        break;
if( j==nr) {
    for (int k=0; k<nr; k++)
        work[k] += alloc[p][k];

    safeseq[count++] = p;
    finish[p] = 1;
    found = true;
  }
 }
}
 if (found== false) {
    printf("system is not in safe state\n"
    return false;
  }
}
 printf("System is in safe state. \n Safe sequ
   is: ");
 for (int i=0; i<np; i++)
     printf("%d", safeSeq[i]);
 printf("\n");
 return true;
}.
int main() {
 int np,nr;
 printf("Enter number of processes: ");
 scanf("%d", &np);
 printf("Enter number of resource types: ");
 scanf("%d", &nr);
 int processes[MAX_process];
 for (int i=0; i<np; i++)
       processes[i] = i;
 int avail[MAX_RESOURCES];
```

```
printf("Enter available resources: ");
for(int i=0; i<m; i++){
  } scanf("%d", &avail[i]);
int max[MAX_PROCESSES][MAX_RESOURCES];
printf("Enter maximum resource matrix:
          \n");
for(int i=0; i<np; i++){
   printf("\nProcess %d: ",i);
   for(int j=0; j<nr; j++){
        scanf("%d", &alloc[i][j]);
   }
~~is=~~ isSafe(processes, avail, max, alloc,
        np, nr);
 return 0;
}
```

Output:

```
Enter the number of process: 5
Enter number of resource: 3
Enter available resource matrix: 3 3 2
Enter maximum resource matrix:
 process 0: 7 5 3
 process 1: 3 2 2
 process 2: 9 0 2
 process 3: 2 2 2
 process 4: 4 3 3

Enter allocation resource matrix:
Process 0: 0 1 0
Process 1: 2 0 0
Process 2: 3 0 2
Process 3: 2 1 1
Process 4: 0 0 2
```

system is in safe state.

safe ~~stack~~ sequence is: 1 3 4 0 2.

3/7/24