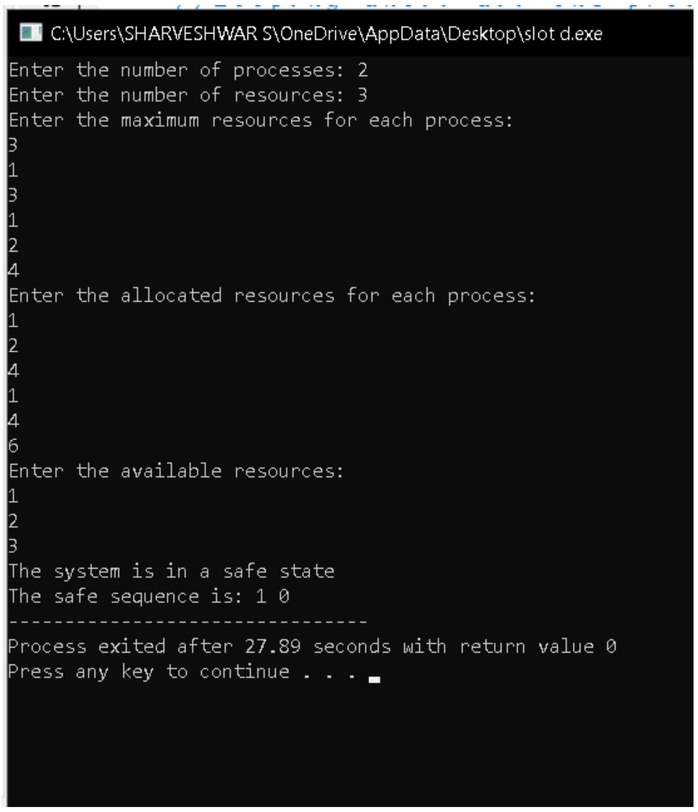
```
#include<stdio.h>

int main()
{
  int i, j, k, m, n, need[10][10], alloc[10][10], max[10][10], avail[10];
  int p[10], sequence[10], sum = 0, count = 0;

  printf("Enter the number of processes: ");
  scanf("%d", &n);
  printf("Enter the number of resources: ");
  scanf("%d", &m);
  printf("Enter the maximum resources for each process: \n");
  for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++) {
      scanf("%d", &max[i][j]);
    }
  }
  printf("Enter the allocated resources for each process: \n");
  for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++) {
      scanf("%d", &alloc[i][j]);
    }
  }
  printf("Enter the available resources: \n");
  for (j = 0; j < m; j++) {
    scanf("%d", &avail[j]);
  }

  //Calculating the need matrix
  for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++) {
        need[i][j] = max[i][j] - alloc[i][j];
    }
}

//Initializing the process to not finished
for (i = 0; i < n; i++) {
    p[i] = 0;
}

//Looping until all the processes are finished
while (count != n) {
    int safe = 0;
    for (i = 0; i < n; i++) {
        if (p[i] == 0) {
            int flag = 0;
            for (j = 0; j < m; j++) {
                if (avail[j] < need[i][j]) {
                    flag = 1;
                    break;
                }
            }
            if (flag == 0) {
                safe = 1;
                for (j = 0; j < m; j++) {
                    avail[j] += alloc[i][j];
                }
                sequence[count++] = i;
                p[i] = 1;
            }
        }
    }
```

```
        if (safe == 0) {
            printf("The system is in an unsafe state\n");
            return 0;
        }
    }
    printf("The system is in a safe state\n");
    printf("The safe sequence is: ");
    for (i = 0; i < n; i++) {
        printf("%d ", sequence[i]);
    }
    return 0;
}
```



```
C:\Users\SHARVESHWAR S\OneDrive\AppData\Desktop\slot d.exe
Enter the number of processes: 2
Enter the number of resources: 3
Enter the maximum resources for each process:
3
1
3
1
2
4
Enter the allocated resources for each process:
1
2
4
1
4
6
Enter the available resources:
1
2
3
The system is in a safe state
The safe sequence is: 1 0
--------------------------------
Process exited after 27.89 seconds with return value 0
Press any key to continue . . .
```