

Program

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

void main() {
    int n,r,i,j,k=0,c=0,flag=0,exec=0;
    printf("Enter the number of processes : ");
    scanf("%d",&n);
    int vis[n];
    int a[n];
    printf("Enter the number of resources : ");
    scanf("%d",&r);
    int alloc[n][r],max[n][r],avail[r],need[n][r];
    printf("Enter the available resources : ");
    for(i=0;i<r;i++)
        scanf("%d",&avail[i]);
    for(i=0;i<n;i++) {
        vis[i]=0;
        printf("Enter the max of the process P%d : ",i);
        for(j=0;j<r;j++)
            scanf("%d",&max[i][j]);
        printf("Enter the allocation of the process P%d : ",i);
        for(j=0;j<r;j++)
            scanf("%d",&alloc[i][j]);
        for(j=0;j<r;j++)
            need[i][j]=max[i][j]-alloc[i][j];
    }
    printf("\nAllocation    Max    Need\n");
    for(i=0;i<n;i++) {
        for(j=0;j<r;j++)
            printf("%d ",alloc[i][j]);
        printf("\t\t");
        for(j=0;j<r;j++)
            printf("%d ",max[i][j]);
        printf("\t");
        for(j=0;j<r;j++)
            printf("%d ",need[i][j]);
        printf("\n");
    }
    while(c<n) {
        flag=0;
        for(i=0;i<n;i++) {
            exec=0;
            for(j=0;j<r;j++) {
                if(need[i][j]<=avail[j])
                    exec++;
            }
            if(exec==r&&vis[i]==0) {
                for(j=0;j<r;j++)
                    avail[j]+=alloc[i][j];
                flag=1;
                vis[i]=1;
                c++;
                a[k++]=i;
            }
        }
        if(flag==0) {
            printf("No safe sequence!!!");
            exit(0);
        }
    }
```

```

}
printf("\nSafe Sequence : \n");
printf("<");
for(i=0;i<n;i++)
    printf("P%d ",a[i]);
printf(">\n");
}

```

Output

```

csea2@sjcet-H81M-DS2:~/maria$ gcc bankers.c
csea2@sjcet-H81M-DS2:~/maria$ ./a.out
Enter the number of processes : 3
Enter the number of resources : 3
Enter the available resources : 3
1
2
Enter the max of the process P0 : 2
1
4
Enter the allocation of the process P0 : 3
2
1
Enter the max of the process P1 : 1
5
3
Enter the allocation of the process P1 : 1
4
2
Enter the max of the process P2 : 3
2
4
Enter the allocation of the process P2 : 12
5
3

Allocation      Max      Need
3 2 1           2 1 4    -1 -1 3
1 4 2           1 5 3     0 1 1
12 5 3          3 2 4    -9 -3 1

Safe Sequence :
<P1 P2 P0 >

```