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

int i,j,no,res;

void OUTPUT(int arr[10][10])
{  printf("\t\t");
  for(i=0;i<res;i++)
    printf("R%d\t",i);

  for(i=0;i<no;i++)
  {  printf("\nP%d\t",i);
    for(j=0;j<res;j++)
      printf("\t%d",arr[i][j]);     }}

int SAFETY(int A[10][10],int W[10],int N[10][10],int seq[])
{  int x=0,flg=0,target=0;
  int finish[10];

  for(i=0;i<no;i++)
    finish[i]=0;

  for(int w=0;w<no;w++)
  {  label:
    for(int i=0;i<no;i++)
    {  flg=0;
      for(int j=0;j<res;j++)
        if(N[i][j] > W[j])
          flg++;
      if(flg==0 && finish[i]==0)
      {  for(j=0;j<res;j++)
          W[j]+=A[i][j];
        finish[i]=1;
        target++;
        seq[x++]=i;
        goto label;      }
      continue;          }}
  if(target==no)
    return 1;
  else
  {  printf("\nThe System is in UNSAFE STATE :(\n");
    return 0;     }}
```

```
void REQUEST(int A[10][10],int N[10][10],int req[10],int AV[10],int ID)
{  int seq[10];
   for(i=0;i<res;i++)
   {  if(req[i] > N[ID][i])
      {  printf("\nNOT POSSIBLE as the process P%d has exceeded its max
claim!!",ID);
         exit(0);      }}

   for(i=0;i<res;i++)
   {  if(req[i] > AV[i])
      {  printf("\nThe process P%d has to wait since resources are not available
yet!!",ID);
         exit(0);   }}

   for(j=0;j<res;j++)
   {  AV[j] -= req[j];
      A[ID][j] += req[j];
      N[ID][j] -= req[j];    }

   int ch=SAFETY(A,AV,N,seq);
   if(ch==1)
   {  printf("\nThe System is in SAFE state. Hence, the resources can be
allocated.\n");
      printf("\nThe Safe Sequence: ");
      for(i=0;i<no;i++)
         printf("|P%d|",seq[i]);       }
   printf("\n");        }

void main()
{  int
ans,id,seq[10],R[10],A[10][10],C[10][10],N[10][10],W[10]={0},req[10],AV[10];

   printf("\nEnter the number of Processes: ");
   scanf("%d",&no);

   printf("Enter the number of Resources: ");
   scanf("%d",&res);

   printf("Enter the max available instances of each resource:\n");
   for(i=0;i<res;i++)
   {  printf("R%d: ",i);
      scanf("%d",&R[i]);     }

   printf("\nEnter the Allocated Resource Table:\n");
   for(i=0;i<no;i++)
      for(j=0;j<res;j++)
         scanf("%d",&A[i][j]);
printf("\nEnter the Maximum Claim Table:\n");
for(i=0;i<no;i++)
  for(j=0;j<res;j++)
    scanf("%d",&C[i][j]);

for(i=0;i<no;i++)
  for(j=0;j<res;j++)
    N[i][j]=C[i][j]-A[i][j];

printf("\nAllocated Resource Table:\n");
OUTPUT(A);
printf("\nMaximum Claim Table:\n");
OUTPUT(C);
printf("\nNeed Matrix:\n");
OUTPUT(N);

for(j=0;j<res;j++)
{  for(i=0;i<no;i++)
    W[j]+=A[i][j];
  W[j]=R[j]-W[j];     }

for(j=0;j<res;j++)
  AV[j]=W[j];

int ch=SAFETY(A,W,N,seq);
if(ch==1)
{  printf("\n\nThe System is in SAFE STATE :)\n");
  printf("\nThe Safe Sequence: ");
  for(i=0;i<no;i++)
    printf("|P%d|",seq[i]);

  printf("\nResource Request for a Process Needed? (1=YES, 0=NO): ");
  scanf("%d",&ans);
  if(ans==1)
  {  printf("Enter the Process ID for initiating request: ");
    scanf("%d",&id);
    printf("Enter the Request Vector for P%d : ",id);
    for(j=0;j<res;j++)
      scanf("%d",&req[j]);
    REQUEST(A,N,req,AV,id);     }}}
```