```
#include <stdio.h>

#define P 5
#define R 3


int main()
{
int MaxMatrix[P][R], needMatrix[P][R], allocationMatrix[P][R], available[R], finished[P];
int i, j, process, count;
count = 0;


for(i = 0; i< P; i++)
        finished[i] = 0;


printf("\n\nEnter the MaxMatrix  for each process : ");
for(i = 0; i < P; i++)
{
        printf("\nP%d : ", i);
        for(j = 0; j < R; j++)
                scanf("%d", &MaxMatrix[i][j]);
}
```

```
printf("\n\nEnter the allocation for each process : ");
for(i = 0; i < P; i++)
{
    printf("\nP%d : ",i);
    for(j = 0; j < R; j++)
        scanf("%d", &allocationMatrix[i][j]);
}

printf("\n\nEnter the Available Resources : ");
for(i = 0; i < R; i++)
        scanf("%d", &available[i]);



    for(i = 0; i < P; i++)
        for(j = 0; j < R; j++)
            needMatrix[i][j] = MaxMatrix[i][j] - allocationMatrix[i][j];

do
{



    process = -1;

    for(i = 0; i < P; i++)
```

```
{
    if(finished[i] == 0)//if not finished
    {
        process = i ;
        for(j = 0; j < R; j++)
        {
            if(available[j] < needMatrix[i][j])
            {
            process = -1;
                break;
            }
        }
    }
    if(process != -1)
        break;
}

if(process != -1)
{
    count++;
    for(j = 0; j < R; j++)
    {
        available[j] += allocationMatrix[process][j];
        allocationMatrix[process][j] = 0;
        MaxMatrix[process][j] = 0;
```

```
            finished[process] = 1;
        }
    }
}while(count != P && process != -1);

if(count == P)
{
    printf("\nThe system is in a safe state!!\n");

}
else
    printf("\nThe system is in an unsafe state!!");}
```