PROGRAM :

```
#include <stdio.h>
#include <conio.h>
int max[100][100];
int alloc[100][100];
int need[100][100];
int avail[100];
int n,r;
void input();
void show();
void cal();
int main()
{
    printf("**********Banker's Algo**********");
    input();
    show();
    cal();
    getch();
    return 0;
}
void input()
{
    int i,j;
    printf("\nEnter no. of processes: ");
    scanf("%d",&n);
    printf("Enter the no. of resource instances: ");
    scanf("%d",&r);
    printf("Enter the max matrix\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<r;j++)
            scanf("%d",&max[i][j]);
    }
    printf("Enter the allocation matrix\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<r;j++)
            scanf("%d",&alloc[i][j]);
    }
    printf("Enter the available matrix\n");
    for(j=0;j<r;j++)
        scanf("%d",&avail[j]);
}

void show()
{
    int i,j;
    printf("Process\tAllocation\tMax\tAvailable\t");
    for(i=0;i<n;i++)
    {
        printf("\nP%d\t\t",i+1);
        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\t");
        if(i==0){
            for(j=0;j<r;j++)
                printf("%d",avail[j]);
        }
    }
}
void cal(){
    int finish[100],temp,need[100][100],flag=1,k,c1=0;
    int safe[100],i,j;
    for(i=0;i<n;i++)
        finish[i]=0;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++)
            need[i][j] = max[i][j]-alloc[i][j];
    }
    printf("\n");
    while(flag){
        flag=0;
        for(i=0;i<n;i++){
            int c=0;
            for(j=0;j<n;j++){
                if(finish[i]==0 && need[i][j]<=avail[j]){
                    c++;
                    if(c==r){
                        for(k=0;k<r;k++){
                            avail[k] += alloc[i][k];
                            finish[i]=1;
                            flag=1;
                        }
                        printf("P%d->",i);
                        if(finish[i]==1)
                            i=n;
                    }
                }
            }
        }
    }
    for(i=0;i<n;i++){
        if(finish[i]==1)
            c1++;
        else
            printf("p%d->",i);
    }
    if(c1==n)
        printf("\nThe system is in safe state");
    else
```

```
        printf("\nProcess are in deadlock\nSystem is in unsafe state!!!");
}
```

OUTPUT:

```
**********Banker's Algo**********
Enter no. of processes: 5
Enter the no. of resource instances: 3
Enter the max matrix
7
5
3
3
2
2
9
0
2
2
2
2
4
3
3
Enter the allocation matrix
0
1
0
2
0
0
3
0
2
2
1
1
0
0
2
Enter the available matrix
3
3
2
Process   Allocation     Max     Available
P1               010      753          332
P2               200      322
P3               302      902
P4               211      222
P5               002      433
P1->P2->P0->P3->P4->
The system is in safe state
```