2) BANKERS ALGORITHM

```
#include<stdio.h>
#include<stdlib.h>
int n,res;

void bankers(int need[n][res],int allocation[n][res],int available[res],int flag[n])
{
int count=0,process[n],temp;
int k=0;
while(count < n)
{
for(int i=0;i<n;++i) // i=1
{
int check=0; //check=1 //to check resources is available
temp = count;
if(flag[i] == 0)
{
for(int j=0;j<res;++j)
{
if(need[i][j] <= available[j])
++check;
}
if(check == res) //all resources are available
{
flag[i]=1;
for(int j=0;j<res;++j)
available[j] += allocation[i][j];
process[k++] = i+1;
++count;
}
}
}
if(temp==count)
++count;
}

for(int i=0;i<n;++i)
{
if(flag[i]!=1)
{
printf("System is not in safe state\n");
exit(0);
}}

printf("System is in safe state and sequence is \n");
```

```
for(int i=0;i<n-1;++i)
{
printf("P%d -> ",process[i]);
}
printf("P%d\n",process[n-1]);
}

int main()
{
printf("Enter the number of processes\n");
scanf("%d",&n);
printf("Enter the number of resources\n");
scanf("%d",&res);

int need[n][res],allocation[n][res],available[res],total[res];

printf("\nEnter the Need Matrix\n");
for(int i=0;i<n;++i)
for(int j=0;j<res;++j)
scanf("%d",&need[i][j]);

printf("\nEnter the Alocation Matrix\n");
for(int i=0;i<n;++i)
for(int j=0;j<res;++j)
scanf("%d",&allocation[i][j]);

printf("\nEnter the available resources\n");
for(int i=0;i<res;++i)
scanf("%d",&available[i]);

int flag[n]; //to mark completed processes
for(int i=0;i<n;++i)
flag[i]=0;

bankers(need,allocation,available,flag);
}
```

OUTPUT

```
faheemshams@Faheems-MacBook-Air System software % ./bankers
Enter the number of processes
5
Enter the number of resources
3

Enter the Need Matrix
7 4 3
1 2 2
6 0 0
0 1 1
4 3 1

Enter the Alocation Matrix
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2

Enter the available resources
3 3 2
System is in safe state and sequence is
P2 -> P4 -> P5 -> P1 -> P3
faheemshams@Faheems-MacBook-Air System software %
```