6. Bankers Algorithm

```
#include<stdio.h>
void main()
{
int
k=0,output[10],d=0,t=0,ins[5],i,avail[5],allocated
[10][5],need[10][5],MAX[10][5],pno,P[10],rz,j,co
unt=0;
printf("\n Enter the no. of resources:");
scanf("%d",&rz);
printf("\n Enter the max instances of each
resources\n");
for(i=0;i<rz;i++)
{
avail[i]==0;
printf("%c=",(i+97));
scanf("%d",&ins[i]);
}
printf("\n Enter no. of processes:");
scanf("%d",&pno);
printf("\n Enter the allocation matrix\n");
for(i=0;i<rz;i++)
printf("%c",(i+97));
printf("\n");
for(i=0;i<pno;i++)
{
P[i]=i;
printf("P[%]",P[i]);
for(j=0;j<rz;j++)
{
scanf("%d",&allocated[i][j]);
avail[j]+=allocated[i][j];
}
}
printf("\n Enter the MAX matrix\n");
for(i=0;i<rz;i++)
{
printf("%c",(i+97));
avail[i]=ins[i]-avail[i];
}
printf("\n");
for(i=0;i<pno;i++)
{
printf("P",i);
for(j=0;j<rz;j++)
scanf("%d",&MAX[i][j]);
}
printf("\n");
A:d=-1;
for(i=0;i<pno;i++)
{
count=0;t=P[i];
for(j=0;j<rz;j++)
{
need[t][j]=MAX[t][j]-allocated[t][j];
if(need[t][j]<=avail[j])
count++;
}
if(count==rz)
{
output[k++]=P[i];
for(j=0;j<rz;j++)
avail[j]+=allocated[t][j];
}
else
P[++d]=P[i];
}
if(d!=-1)
{
pno=d+1;
goto A;
}
printf("\t<");
for(i=0;i<k;i++)
printf("P[%d]",output[i]);
printf(">");
}
```

OUTPUT

```
Enter the no. of resources:1
 Enter the max instances of each resources
a=20
 Enter no. of processes:5
 Enter the allocation matrix
a
 P[0]2
 P[1]6
 P[2]8
 P[3]4
 P[4]4

  Enter the MAX matrix
  a
  P[0]20
  P[1]18
  P[2]2
  P[3]12
  P[4]6

        <P[2]P[4]P[3]P[1]P[0]>
```