BANKER’s Algorithm Program

```
#include<stdio.h>


int main()
{
	int p,r,i,j,flag;
	int avail_r[10];
	int allocated_r[10][20];
	int max_r[10][20];
	int need[10][20];

	printf("Enter no of processes:");
	scanf("%d",&p);
	printf("Enter no of resources:");
	scanf("%d",&r);

	printf("Available resoruces:\n");
	for(j=0;j<r;j++)
	{

		{
			printf("Enter data in [%d]: ",j);
			scanf("%d",&avail_r[j]);
		}
	}


	printf("Display Array:\n");
	for(i=0;i<r;i++)
	{


		printf("%d\t",avail_r[i]);


		printf("\n");
	}


	printf("Allocated resoruces:\n");
	for(i=0;i<p;i++)
	{
		for(j=0;j<r;j++)
		{
			printf("Enter data in [%d][%d]: ",i,j);
			scanf("%d",&allocated_r[i][j]);
		}
	}

	printf("Display Matrix:\n");
```

```
for(i=0;i<p;i++)
{
	for(j=0;j<r;j++)
	{
		printf("%d\t",allocated_r[i][j]);

	}
	printf("\n");
}

printf("Max resoruces:\n");
for(i=0;i<p;i++)
{
	for(j=0;j<r;j++)
	{
		printf("Enter data in [%d][%d]: ",i,j);
		scanf("%d",&max_r[i][j]);
	}
}

printf("Display Matrix:\n");
for(i=0;i<p;i++)
{
	for(j=0;j<r;j++)
	{
		printf("%d\t",max_r[i][j]);

	}
	printf("\n");
}

printf("Need matrix:\n");
for(i=0;i<p;i++)
{
	for(j=0;j<r;j++)
	{
		need[i][j]=max_r[i][j]-allocated_r[i][j];
		printf("%d\t",need[i][j]);

	}
}

int exe[10];
for(i=0;i<p;i++)
{
	exe[i]=0;
}

while(1)
{
	for(i=0;i<p;i++)
	{
```

```
                        if(exe[i]==0)
                        {
                                flag=1;
                                for(j=0;j<r;j++)
                                {
                                        if(avail_r[j]<need[i][j])
                                        {
                                                flag=0;
                                                break;
                                        }
                                }
                                if(flag==1)
                                {
                                        printf("\n %d is running\n",i);
                                        exe[i]=1;
                                        for(j=0;j<r;j++)
                                        {
                                                avail_r[j]+=allocated_r[i][j];
                                        }
                                        break;
                                }
                        }
                }
                if(i==p)
                {
                        flag=1;
                        for(i=0;i<p;i++)
                        {
                                if(exe[i]==0)
                                {
                                        flag=0;
                                        break;
                                }
                        }
                        if(flag==1)
                        {
                                printf("Safe state");
                        }
                        else
                        {
                                printf("Not safe");
                        }
                        break;
                }

        }
        return 0;
}
```

OUTPUT:

```
pl-lab@pllab-OptiPlex-3000:~$ gcc banker.c
pl-lab@pllab-OptiPlex-3000:~$ ./a.out
```

Enter no of processes:4
Enter no of resources:3
Available resoruces:
Enter data in [0]: 2
Enter data in [1]: 5
Enter data in [2]: 3
Display Array:
2
5
3
Allocated resoruces:
Enter data in [0][0]: 2
Enter data in [0][1]: 4
Enter data in [0][2]: 6
Enter data in [1][0]: 5
Enter data in [1][1]: 7
Enter data in [1][2]: 8
Enter data in [2][0]: 6
Enter data in [2][1]: 8
Enter data in [2][2]: 6
Enter data in [3][0]: 4
Enter data in [3][1]: 3
Enter data in [3][2]: 5
Display Matrix:
2 4 6
5 7 8
6 8 6
4 3 5
Max resoruces:
Enter data in [0][0]: 2
Enter data in [0][1]: 3
Enter data in [0][2]: 5
Enter data in [1][0]: 6
Enter data in [1][1]: 2
Enter data in [1][2]: 6
Enter data in [2][0]: 8
Enter data in [2][1]: 4
Enter data in [2][2]: 2
Enter data in [3][0]: 1
Enter data in [3][1]: 2
Enter data in [3][2]: 4
Display Matrix:
2 3 5
6 2 6
8 4 2
1 2 4
Need matrix:
0 -1 -1 1 -5 -2 2 -4 -4 -3 -1 -1
0 is running

1 is running

2 is running

3 is running
Safe state