Assignment No.5

Implement the C program for Deadlock Avoidance Algorithm: Bankers Algorithm.

5_banker.c

```
#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_r[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_r[i][j]=max_r[i][j]-allocated_r[i][j];
                printf("%d\t",need_r[i][j]);

        }
        printf("\n");
}
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_r[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-17@pl17-OptiPlex-3020:~/IT/07$ gcc 5_banker.c
pl-17@pl17-OptiPlex-3020:~/IT/07$ ./a.out
Enter no of processes:5
Enter no of resources:3
Available resoruces:
```

```
Enter data in [0]: 10
Enter data in [1]: 5
Enter data in [2]: 7
Display Array:
10
5
7
Allocated resoruces:
Enter data in [0][0]: 0
Enter data in [0][1]: 1
Enter data in [0][2]: 0
Enter data in [1][0]: 2
Enter data in [1][1]: 0
Enter data in [1][2]: 0
Enter data in [2][0]: 3
Enter data in [2][1]: 0
Enter data in [2][2]: 2
Enter data in [3][0]: 2
Enter data in [3][1]: 1
Enter data in [3][2]: 1
Enter data in [4][0]: 0
Enter data in [4][1]: 0
Enter data in [4][2]: 2
Display Matrix:
0       1       0
2       0       0
3       0       2
2       1       1
0       0       2
Max resoruces:
Enter data in [0][0]: 7
Enter data in [0][1]: 5
Enter data in [0][2]: 3
Enter data in [1][0]: 3
Enter data in [1][1]: 2
Enter data in [1][2]: 2
Enter data in [2][0]: 9
Enter data in [2][1]: 0
Enter data in [2][2]: 2
Enter data in [3][0]: 4
Enter data in [3][1]: 2
Enter data in [3][2]: 2
Enter data in [4][0]: 5
Enter data in [4][1]: 3
Enter data in [4][2]: 3
Display Matrix:
7       5       3
3       2       2
9       0       2
4       2       2
5       3       3
Need matrix:
```

```
7      4      3
1      2      2
6      0      0
2      1      1
5      3      1

 0 is running

 1 is running

 2 is running

 3 is running

 4 is running
Safe state
```