

Lab 12

Mahammadayan S. Shaikh

Roll Number :- CE121

- **Aim :-** To implement program of Banker's Algorithm.
- **Code :-**

```
#include <stdio.h>
```

```
void bankersAlgo(int p,int r,int available[r],int max[p][r],int allocation[p][r],int  
need[p][r]){
```

```
    int i,j;  
    for(i=0;i<p;i++)  
        for(j=0;j<r;j++)  
            need[i][j]=max[i][j]-allocation[i][j];
```

```
    int isFinish[p],safeSequence[p],work[r];
```

```
    for(i=0;i<p;i++)  
        isFinish[i]=0;
```

```
    for(i=0;i<r;i++)  
        work[i]=available[i];
```

```
    int cnt=0;  
    while(cnt<p){  
        int procFound=0;  
        for(j=0;j<p;j++){  
            if(isFinish[j])  
                continue;  
            int k;  
            for(k=0;k<r;k++)
```

```

        if(need[j][k]>work[k])
            break;
    if(k==r){
        for(int l=0;l<r;l++)
            work[l]+=allocation[j][l];

        safeSequence[cnt++]=j;
        isFinish[j]=1;
        procFound=1;
    }
}
if(!procFound){
    printf("No Safe Sequence is possible\n");
    return;
}
}
printf("Safe Sequence : ");
for(i=0;i<p;i++)
    printf("%d ",safeSequence[i]);
printf("\n");
}

```

```

void main(){
    int i,j,r,p; // r-> #resources p-> #processes
    printf("Enter number of resources : ");
    scanf("%d",&r);
    int available[r];
    printf("Enter available resoruces : ");
    for(i=0;i<r;i++)
        scanf("%d",&available[i]);
    printf("Enter number of processes : ");
    scanf("%d",&p);
    int max[p][r],allocation[p][r],need[p][r];

    printf("Enter max matrix : \n");
}

```

```

    for(i=0;i<p;i++)
        for(j=0;j<r;j++)
            scanf("%d",&max[i][j]);

    printf("Enter allocation matrix : \n");
    for(i=0;i<p;i++)
        for(j=0;j<r;j++)
            scanf("%d",&allocation[i][j]);

    bankersAlgo(p,r,available,max,allocation,need);

}

```

- **Output :-**

```

ayan@ayan-Lenovo-ideapad-130-15IKB: ~/sem5/OS/Lab12
ayan@ayan-Lenovo-ideapad-130-15IKB:~/sem5/OS/Lab12$ gcc Bankers-Algorithm.c
ayan@ayan-Lenovo-ideapad-130-15IKB:~/sem5/OS/Lab12$ ./a.out
Enter number of resources : 4
Enter available resoruces : 1 5 2 0
Enter number of processes : 5
Enter max matrix :
0 2 1 0
1 6 5 2
2 3 6 6
0 6 5 2
0 6 5 6
Enter allocation matrix :
0 1 1 0
1 2 3 1
1 3 6 5
0 6 3 2
0 0 1 4
Safe Sequence : 0 3 4 1 2
ayan@ayan-Lenovo-ideapad-130-15IKB:~/sem5/OS/Lab12$

```

Test Case 1

```
ayan@ayan-Lenovo-ideapad-130-15IKB: ~/sem5/OS/Lab12
ayan@ayan-Lenovo-ideapad-130-15IKB:~/sem5/OS/Lab12$ gcc Bankers-Algorithm.c
ayan@ayan-Lenovo-ideapad-130-15IKB:~/sem5/OS/Lab12$ ./a.out
Enter number of resources : 3
Enter available resoruces : 3 3 2
Enter number of processes : 5
Enter max matrix :
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter allocation matrix :
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Safe Sequence : 1 3 4 0 2
ayan@ayan-Lenovo-ideapad-130-15IKB:~/sem5/OS/Lab12$
```

Test Case : 2