



Academy of  
Engineering

## Operating Systems Lab

### Banker's Algorithm

Submitted By:

**Fiza Patel**

**Roll No: 191**

**PRN No: 0120190425**

**Branch and Batch: TY C3 (IT)**

## Program:

```
#include <iostream>

using namespace std;

int main()
{
    int nos_of_proc, nos_of_res, i, j;

    cout<<"-----Banker's Algorithm-----";
    cout<<"\nEnter Number of Processes: ";
    cin>>nos_of_proc;
    cout<<"Enter number of Resources: ";
    cin>>nos_of_res;

    int allocated[nos_of_proc][nos_of_res],
    max[nos_of_proc][nos_of_res], available[nos_of_res],
    requestCheck[nos_of_res], request[nos_of_res];

    cout<<"Enter elements of allocated matrix: \n";
    for(i=0;i<nos_of_proc;i++)
    {
        for(j=0;j<nos_of_res;j++)
        {
            cin>>allocated[i][j];
        }
        cout<<"\n";
    }

    cout<<"Enter elements of max matrix: \n";
```

```

for(i=0;i<nos_of_proc;i++)
{
    for(j=0;j<nos_of_res;j++)
    {
        cin>>max[i][j];
    }
    cout<<"\n";
}
cout<<"Enter elements of available 1-D matrix: \n";
for(i=0;i<nos_of_res;i++)
{
    cin>>available[i];
    requestCheck[i] = available[i];
}
int need[nos_of_proc][nos_of_res];
for(i=0;i<nos_of_proc;i++)
{
    for(j=0;j<nos_of_res;j++)
    {
        need[i][j] = max[i][j] - allocated[i][j];
    }
}
int f[nos_of_proc];
for(i=0;i<nos_of_proc;i++)
{

```

```

    f[i]=0;
}

int count=nos_of_proc, final[nos_of_proc], ind = 0;
do
{
    for(i=0;i<nos_of_proc;i++)
    {
        int flag=0;
        for(j=0;j<nos_of_res;j++)
        {
            if(need[i][j]>available[j])
            {
                flag=1;
                break;
            }
        }
        int mark = 0;
        if(flag==0)
        {
            if(f[i]==0)
            {
                mark=1;
            }
            f[i]=1;
            for(j=0;j<nos_of_res;j++)

```

```

        {
            available[j]+=allocated[i][j];
        }
        if(mark==1)
        {
            final[ind++]=i;
        }
    }
}

count=count-1;
if(ind==nos_of_proc)
{
    break;
}
}while(count>0);
if(ind==nos_of_proc)
{
    cout<<"\nThe safe sequence is: ";
    for(int k=0;k<nos_of_proc;k++)
    {
        cout<<"P"<<final[k]<<"->";
    }
}
else{
    cout<<"\nIt is in unsafe";

```

```

    }

    cout<<"\nEnter your request for a process: \n";
    for(i=0;i<nos_of_res;i++)
    {
        cin>>request[i];
    }

    int flag=0;
    for(j=0;j<nos_of_res;j++)
    {
        if(request[j]>requestCheck[j])
        {
            flag=1;
            cout<<"\nRequest Can't be fulfilled\n";
            break;
        }
    }

    if(flag==0)
    {
        cout<<"\nRequest is fulfilled\n";
    }

    return 0;
}

```

## Output:

### Allocation Matrix:

```
E:\Tests\banker c++.exe
-----Banker's Algorithm-----
Enter Number of Processes: 5
Enter number of Resources: 3
Enter elements of allocated matrix:
0
1
0

2
0
0

3
0
2

2
1
1

0
0
2
```

### Max Matrix and 1-D Available resources Matrix:

```
E:\Tests\banker c++.exe
Enter elements of max matrix:
7
5
3

3
2
2

9
0
2

2
2
2

4
3
3

Enter elements of available 1-D matrix:
3
3
2
```

## Safe Sequence and Request Status:

```
E:\Tests\banker c++.exe
Enter elements of available 1-D matrix:
3
3
2

The safe sequence is: P1->P3->P4->P0->P2->
Enter your request for a process:
2
1
1

Request is fulfilled

-----
Process exited after 54.82 seconds with return value 0
Press any key to continue . . .
```