

## AIM Priority

```
import java.util.Scanner;

public class Bankers{

private int need[][][],allocate[][][],max[][][],avail[][][],np,nr;

private void input(){

Scanner sc=new Scanner(System.in);

System.out.print("Enter no. of processes and resources :");

np=sc.nextInt(); //no. of process

nr=sc.nextInt(); //no. of resources

need=new int[np][nr]; //initializing arrays

max=new int[np][nr];

allocate=new int[np][nr];

avail=new int[1][nr];

System.out.println("Enter allocation matrix -->");

for(int i=0;i<np;i++)

for(int j=0;j<nr;j++)

allocate[i][j]=sc.nextInt(); //allocation matrix

System.out.println("Enter max matrix -->");

for(int i=0;i<np;i++)

for(int j=0;j<nr;j++)

max[i][j]=sc.nextInt(); //max matrix

System.out.println("Enter available matrix -->");

for(int j=0;j<nr;j++)

avail[0][j]=sc.nextInt(); //available matrix

sc.close();

}

private int[][] calc_need(){

for(int i=0;i<np;i++)
```

```

for(int j=0;j<nr;j++) //calculating need matrix
need[i][j]=max[i][j]-allocate[i][j];
return need;
}

private boolean check(int i){
//checking if all resources for ith process can be allocated
for(int j=0;j<nr;j++)
if(avail[0][j]<need[i][j])
return false;
return true;
}

public void isSafe(){
input();
calc_need();
boolean done[] = new boolean[np];
int j=0;while(j<np){ //until all process allocated
boolean allocated=false;
for(int i=0;i<np;i++)
if(!done[i] && check(i)){ //trying to allocate
for(int k=0;k<nr;k++)
avail[0][k]=avail[0][k]-need[i][k]+max[i][k];
System.out.println("Allocated process : "+i);
allocated=done[i]=true;
}
if(!allocated) break; //if no allocation
}
if(j==np) //if all processes are allocated
System.out.println("\nSafely allocated");
}

```

```
else
System.out.println("All process cant be allocated safely");
}
public static void main(String[] args) {
new Bankers().isSafe();
}
}
```