	The state of the s						
	C. D. was Algorithm						
	9. Bankers Algorithm						
	a have desither for deadlock detection						
	Aim: - Bankers Algonium av de de						
	Aim: - Bankers Algorithm for deadlock altection and avoidance						
	Problem Statement: - write a Java program to implement Banker's Algorithm:						
	implement barrers agontim.						
	Theory:						
	fully Data securities are used to implement						
	the Bankeris Algorom:						
1.30	The built and highwin.						
(P)	Available :-						
0	Marx						
(3)	Meed						
	meed 3 3 A						
	The few sections and the section of						
	Need [i,j] = Max [i,j] - Allocation [i,j]						
*	Safety Algorithm:						
)	Let work and finish be wectors of length 'm's						
	o n'sespectively.						
	Initialize: Work = Available						
	Finish [i] = false; for i=1,2,3,4						

prind an i such that both a) finish [i] = false 5) Need; < = West if no such i rists goto step (4) 5) West = West + Allocation (i) finish [i] = true goto step (2) 4) if finish (i] = true for all i then the system is in a safe state. auestion 1: What will be Content of Need matrix Need [i, j] = Max [i, j] - Allocation [i, j].									
a) finsh (i) = two step (4) if no such i exists go to step (4) s) NORK = NORK + Allocation (i) finish (i) = true go to step (2) 4) if finish (i) = true for all i then the system is in a safe state. augston 1: What will be content of Need matrix									
a) finish (i) = two step (4) if no such i exists go to step (4) s) week = werk + Allocation (i) finish (i) = true go to step (2) 4) if finish (i) = true for all i then the system is in a safe state. and then I: what will be content of weed matrix	1		- h +H	hat b	et 4	RESERVED TO THE RESERVED OF THE PERSON OF TH			
5) Need; <= West go to step (4) if no such i rists go to step (4) 5) Wesk = Wesk + Allocation (i) Finish [i] = true go to step (2) 4) if finish (i) = true for all i then the system is in a safe state. auestion 1: What will be content of Need matrix	2)	a) finish CIJ = raise							
s) work = work + Allocation (i) Finish [i] = true goto step (2) 4) if finish (i) = true for all i then the system is in a safe state. aucstion 1: what will be content of Need matrix									
3) WORK = WORK + Allocation (i) 6 nish (i] = true goto step (2) 4) if finish (i] = true for all i then the system is in a safe state. auastion 1: What will be content of Need matrix		if no such i exists goto step (4)							
goto step (2) 4) if finish (i) = true for all i then the system is in a safe state. aucstion 1: what will be content of Need matrix									
goto step (2) 4) if finish (i) = true for all i then the system is in a safe state. aucstion 1: what will be content of Need matrix	5)	WORK = WORK + Allocation (i)							
4) if finish (i) = true for all i then the system is in a safe state. auestion 1: what will be content of Need matrix		Engli [i] = true							
4) if finish (i) = true for all i then the system is in a safe state. auestion 1: what will be content of Need matrix		goto step (2)							
auestion 1: what will be content of need makix	_								
auestion 1: what will be content of need makix	4)	if finish (i) = true for all i							
		then the system is in a safe state.							
		TOURNESS OF ENGLISHED PROPERTY ALL CONTROL OF THE PROPERTY OF							
· Need [i, j] = Max [i, j] - Allocation [i, j] .									
Process Heed		Process		A STATE OF THE PARTY OF THE PAR		(8 Attacasto)			
A B C				B	<u>C</u>	1.9014 (4)			
P, 1 2 2 4			+	4					
P, 6 0 0			6	A CONTRACTOR OF THE PARTY OF TH		STATE OF THE STATE			
P ₃ 0 1 1				1		Filmin william			
P4 4 3 1		Pu		3	1				
Total was at the Anna to day that a		La Bour Cong	and a second	-1	Lung	Lo Diane Hot 19			
aller the page of the contract									

Assignment No. 09

Problem Satement: Write a Java program to implement Banker's Algorithm

1. Banker's Algorithm Program:

```
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; j++; } if(!allocated)
break; //if no allocation
} if(j==np) //if all processes are
allocated System.out.println("\nSafely
allocated"); else
System.out.println("All proceess cant be allocated safely");
}
public static void main(String[] args) {
new Bankers().isSafe();
}
```

```
Enter no. of processes and resources : 4
Enter allocation matrix -->
Enter allocation matro

1
0
2
0
3
0
2
2
1
1
Enter max matrix -->
7
Enter max matrix -->
7
5
3
3
2
2
9
0
2
2
2
Enter available matrix -->
3
2
Allocated process : 1
Allocated process : 3
Allocated process : 0
Allocated process : 2
Safely allocated
```