

NAME – PINKY PAMECHA

ROLL NO-C114

SAP ID-60004220056

## EXPERIMENT NO 5

### BANKER'S ALGORITHM

CODE:

```
import java.util.Scanner;

public class Main{
    int n; // Number of processes
    int m; // Number of resources
    int need[][];
    int[][] max;
    int[][] alloc;
    int[] avail;
    int safeSequence[];
    void isSafe()
    {
        int count=0;

        //visited array to find the already allocated process
        boolean visited[] = new boolean[n];
        for (int i = 0;i < n; i++)
        {
            visited[i] = false;
        }

        //work array to store the copy of available resources
        int work[] = new int[m];
        for (int i = 0;i < m; i++)
        {
            work[i] = avail[i];
        }

        while (count<n)
        {
            boolean flag = false;
            for (int i = 0;i < n; i++)
            {
                if (visited[i] == false)
                {
                    int j;
                    for (j = 0;j < m; j++)
                    {
                        if (need[i][j] > work[j])
                            break;
                    }
                }
            }
        }
    }
}
```

```

        }
        if (j == m)
        {
            safeSequence[count++]=i;
            visited[i]=true;
            flag=true;

            for (j = 0;j < m; j++)
            {
                work[j] = work[j]+alloc[i][j];
            }
        }
    }
    if (flag == false)
    {
        break;
    }
}
if (count < n)
{
    System.out.println("The System is UnSafe!");
}
else
{
    //System.out.println("The given System is Safe");
    System.out.println("Following is the SAFE Sequence");
    for (int i = 0;i < n; i++)
    {
        System.out.print("P" + safeSequence[i]);
        if (i != n-1)
            System.out.print(" -> ");
    }
}

void calculateNeed()
{
    for (int i = 0;i < n; i++)
    {
        for (int j = 0;j < m; j++)
        {
            need[i][j] = max[i][j]-alloc[i][j];
        }
    }
}

void initializeValues() {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the number of processes: ");

```

```

n = scanner.nextInt();

System.out.print("Enter the number of resources: ");
m = scanner.nextInt();

need = new int[n][m];
max = new int[n][m];
alloc = new int[n][m];
avail = new int[m];
safeSequence = new int[n];

// Allocation Matrix
System.out.println("Enter the Allocation Matrix:");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        alloc[i][j] = scanner.nextInt();
    }
}

// MAX Matrix
System.out.println("Enter the MAX Matrix:");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        max[i][j] = scanner.nextInt();
    }
}

// Available Resources
System.out.println("Enter the Available Resources:");
for (int i = 0; i < m; i++) {
    avail[i] = scanner.nextInt();
}

// The rest of your code remains the same

public static void main(String[] args) {
    Main gfg = new Main();
    System.out.println("PINKY PAMECHA C114 60004220056");
    gfg.initializeValues();
    // Calculate the Need Matrix
    gfg.calculateNeed();

    // Check whether the system is in a safe state or not
    gfg.isSafe();
}
}

```

OUTPUT:

PINKY PAMECHA C114 60004220056

Enter the number of processes: 5

Enter the number of resources: 4

Enter the Allocation Matrix:

0 1 1 0

1 2 3 1

1 3 6 5

0 6 3 2

0 0 1 4

Enter the MAX Matrix:

0 2 1 0

1 6 5 2

2 3 6 6

0 6 5 2

0 6 5 6

Enter the Available Resources:

1 5 2 0

Following is the SAFE Sequence

P0 -> P3 -> P4 -> P1 -> P2

...Program finished with exit code 0

Press ENTER to exit console.