

Assignment 05

Roll No. 33331

Program :

//Topic : Banker's Algorithm

```
public class Main{
```

```
    static int n = 5, m = 3;
```

```
    public static void main(String[] args){
```

```
        //resource A=10,B=5,C=7
```

```
        int[][] maxNeed = {
```

```
            {7,5,3}, //P1
```

```
            {3,2,2}, //P2
```

```
            {9,0,2}, //P3
```

```
            {4,2,2}, //P4
```

```
            {5,3,3} //P5
```

```
        };
```

```
        int[][] aloc = {
```

```
            //random allocation from us
```

```
            {2,1,0},
```

```
            {2,0,0},
```

```
            {3,0,2},
```

```
            {2,1,1},
```

```
            {0,0,2}
```

```
            // 7,2,5
```

```
        };
```

```

//total available - total allocated
int[] avail = {3,3,2}; //10-7, 5-2, 7-5
int[][] remNeed = new int[n][m];
//filling remNeed array here
for(int i=0; i<n;i++){
    for(int j= 0; j<m;j++){
        remNeed[i][j] = maxNeed[i][j] - alloc[i][j];
    }
}
System.out.println("Max Need Resources Array : ");
printArr(maxNeed);
System.out.println("Allocated resources Array : ");
printArr(alloc);
System.out.println("Remaining Need Array : ");
printArr(remNeed);
int[] vis = new int[n];
int[] ans = new int[n];
int count = 0;
int idx = 0;
while(count < 5){
    idx = idx % 5;
    if(vis[idx] == 0 && avail[0] >= remNeed[idx][0] && avail[1] >= remNeed[idx][1] &&
avail[2]
    >= remNeed[idx][2]){
        vis[idx] = 1;
        count++;

        System.out.println("Available resources for P"+(idx+1)+" : "+avail[0]+" "+avail[1]+"
"+avail[2]);

```

```

        avail[0] += alloc[idx][0];
        avail[1] += alloc[idx][1];
        avail[2] += alloc[idx][2];
        ans[count-1] = idx;
    }
    idx++;
}
System.out.println();
for(int i=0;i<n;i++){
    System.out.print("P"+(i+1)+" -> ");
}
System.out.println("Done!!!");
}

```

```

private static void printArr(int[][] arr){
    for(int i=0; i<n;i++){
        System.out.print("P"+(i+1)+" ");
        for(int j= 0; j<m;j++){
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
    System.out.println();
}
}

```

Output :

java -cp /tmp/e5TT4E0ZBr/Main

Max Need Resources Array :

P1 7 5 3

P2 3 2 2

P3 9 0 2

P4 4 2 2

P5 5 3 3

Allocated resources Array :

P1 2 1 0

P2 2 0 0

P3 3 0 2

P4 2 1 1

P5 0 0 2

Remaining Need Array :

P1 5 4 3

P2 1 2 2

P3 6 0 0

P4 2 1 1

P5 5 3 1

Available resources for P2 : 3 3 2

Available resources for P4 : 5 3 2

Available resources for P5 : 7 4 3

Available resources for P1 : 7 4 5

Available resources for P3 : 9 5 5

P1 -> P2 -> P3 -> P4 -> P5 -> Done!!!

=== Code Execution Successful ===