Practical no: 11

Problem Statement: - Study assignment on process scheduling algorithms in Android and Tizen.

Name: Aditi Dinesh Mulay

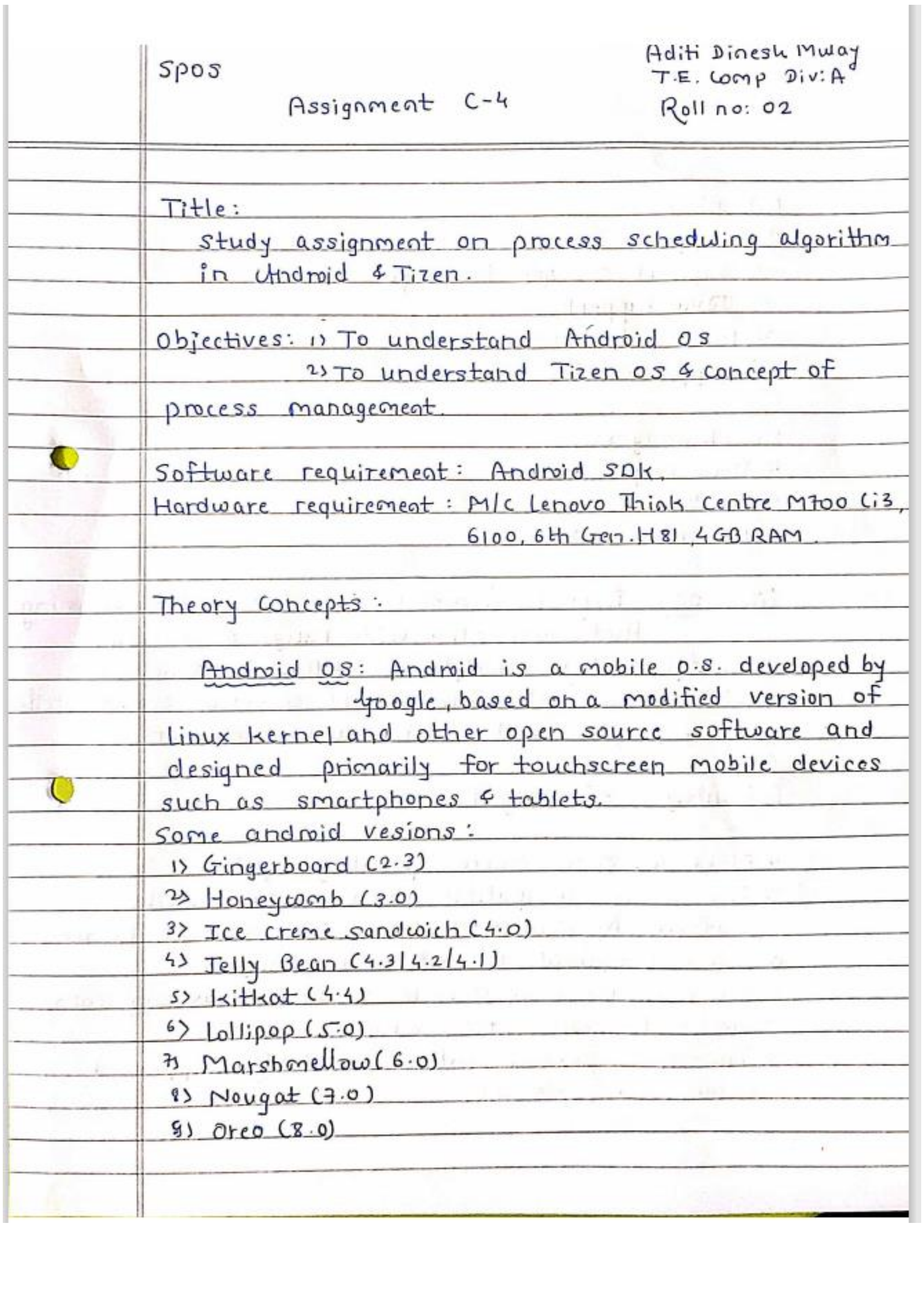
Class: T.E. Computer

Subject: SPOS

Div: A

Roll no: 02

PRN No. 71918146B



SPOS

Assignment C-4

Aditi Dinesh Muway
T.E. Comp Div: A
Roll no: 02

Title:
study assignment on process scheduling algorithm in Android & Tizen.

Objectives: 1) To understand Android OS
2) To understand Tizen OS & concept of process management.

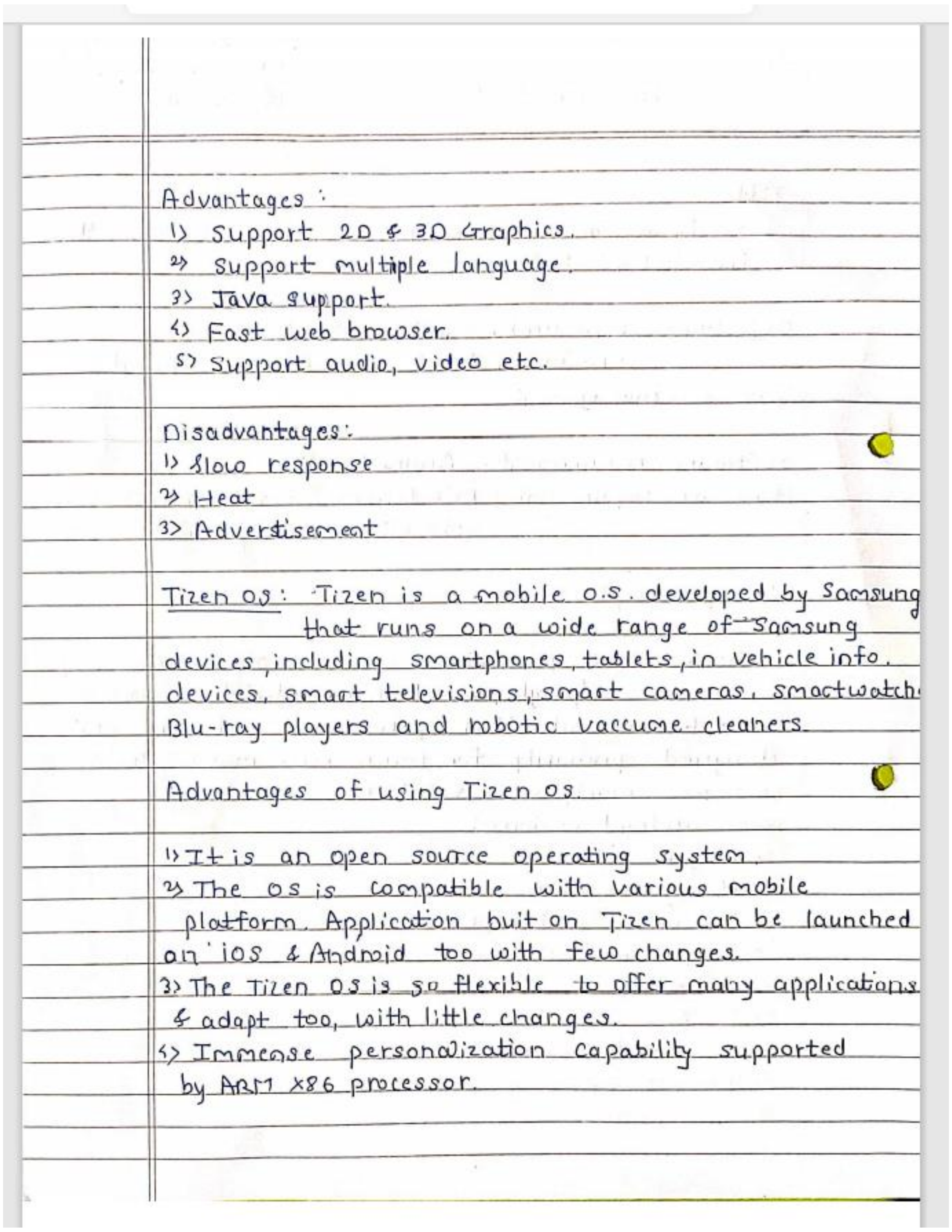
Software requirement: Android SDK.
Hardware requirement: M/C Lenovo Think Centre M700 Ci3, 6100, 6th Gen. H81, 4GB RAM.

Theory concepts:

Android OS: Android is a mobile O.S. developed by Google, based on a modified version of Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones & tablets.

Some android vesions:

1) Gingerboard (2.3)
2) Honeycomb (3.0)
3) Ice creme sandwich (4.0)
4) Jelly Bean (4.3/4.2/4.1)
5) Kitkat (4.4)
6) Lollipop (5.0)
7) Marshmellow (6.0)
8) Nougat (7.0)
9) Oreo (8.0)



Advantages :

1) Support 2D & 3D Graphics.
2) Support multiple language.
3) Java support.
4) Fast web browser.
5) Support audio, video etc.

Disadvantages :

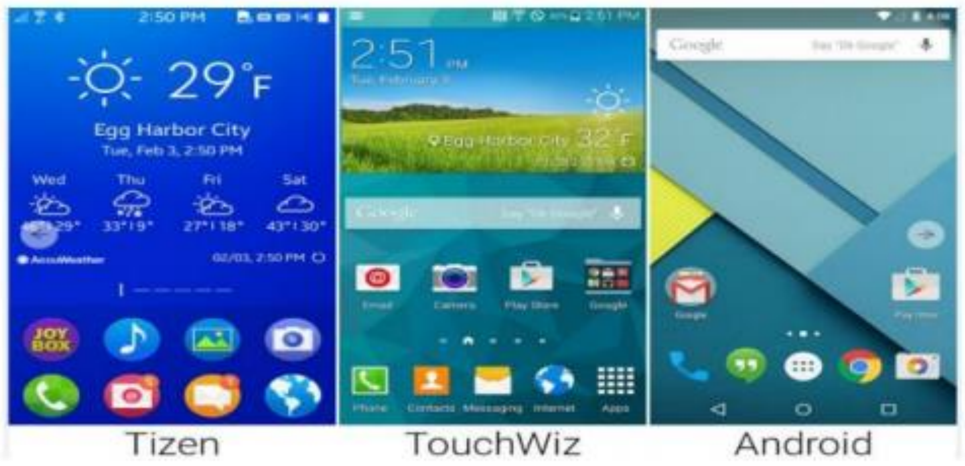
1) Slow response
2) Heat
3) Advertisement

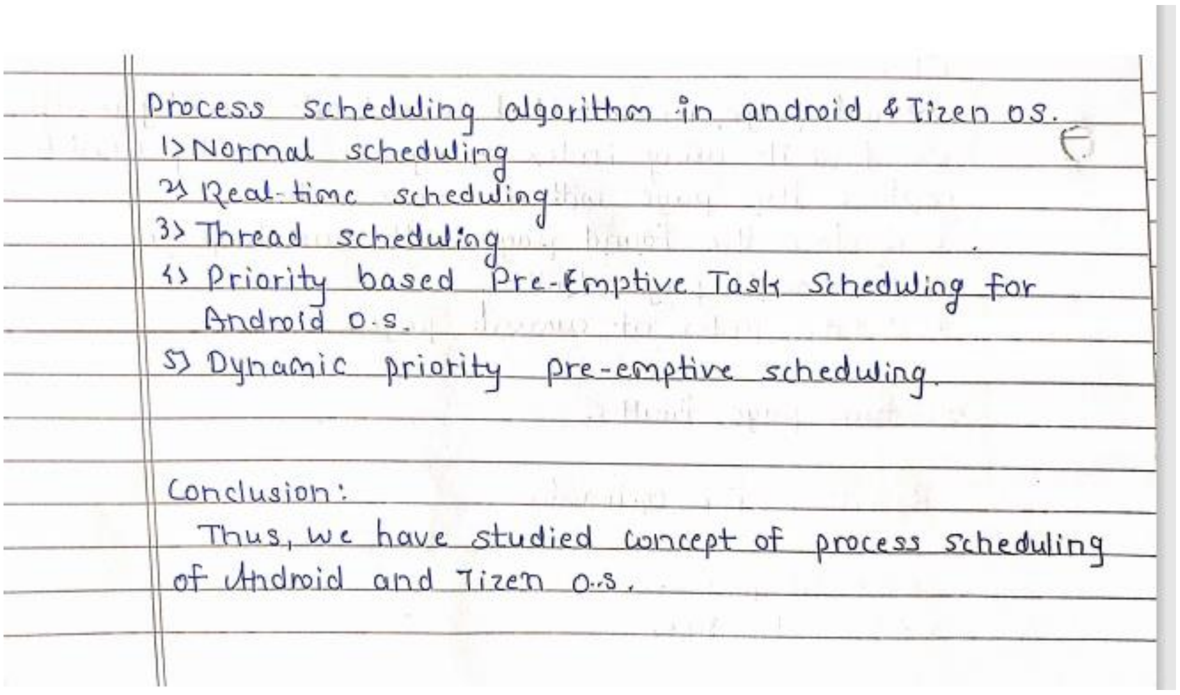
Tizen OS : Tizen is a mobile O.S. developed by Samsung that runs on a wide range of Samsung devices, including smartphones, tablets, in vehicle info. devices, smart televisions, smart cameras, smartwatch Blu-ray players and robotic vaccume cleaners.

Advantages of using Tizen OS.

1) It is an open source operating system.
2) The OS is compatible with various mobile platform. Application buit on Tizen can be launched on iOS & Android too with few changes.
3) The Tizen OS is so flexible to offer many applications & adapt too, with little changes.
4) Immense personalization capability supported by ARM X86 processor.

Android vs Tizen Operating system :





Process scheduling algorithm in android & Tizen OS.

1> Normal scheduling

2> Real-time scheduling

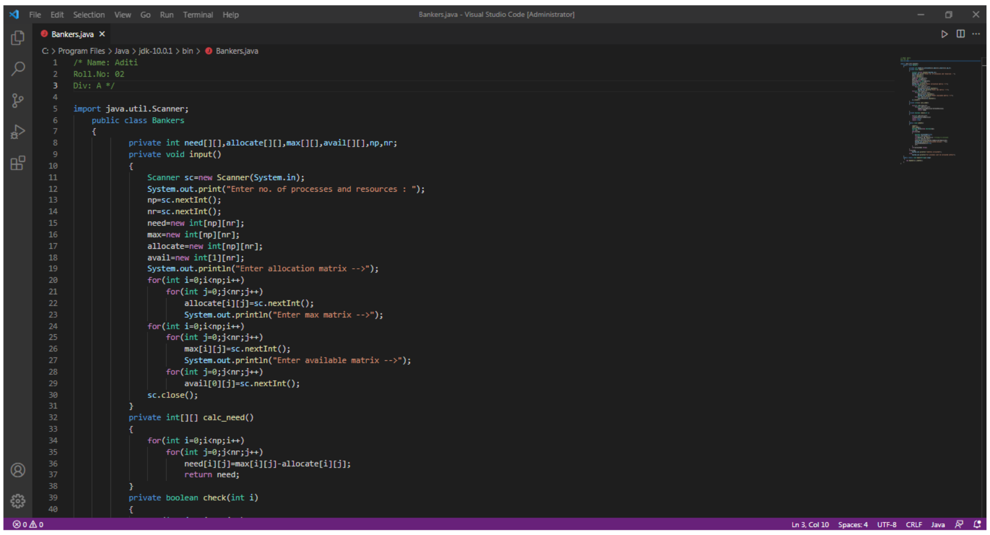
3> Thread scheduling

4> Priority based Pre-Emptive Task Scheduling for Android O.S.

5> Dynamic priority pre-emptive scheduling.

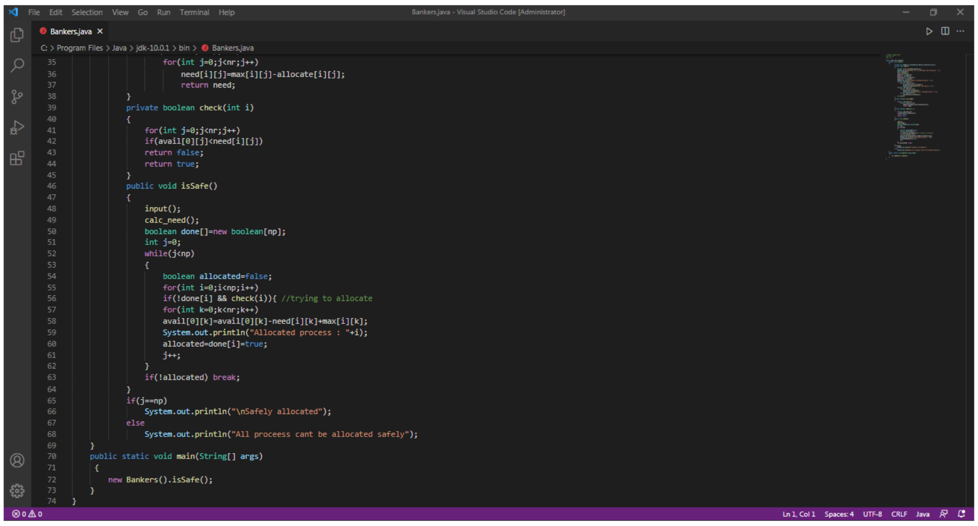
Conclusion:

Thus, we have studied concept of process scheduling of Android and Tizen O.S.

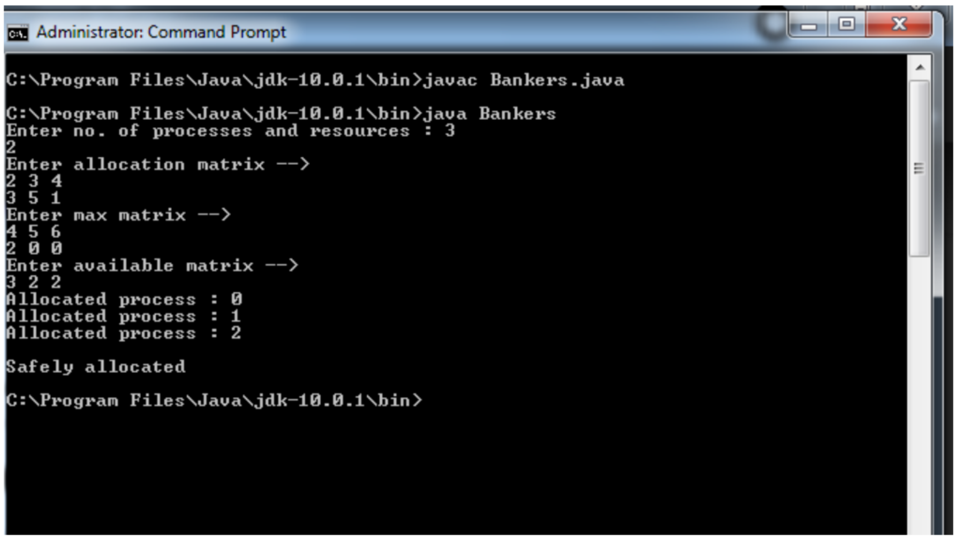

Bankers.java - Visual Studio Code [Administrator]
Bankers.java
C: > Program Files > Java > jdk-10.0.1 > bin > Bankers.java
Ln 3, Col 10 Spaces: 4 UTF-8 CRLF Java

```java
/* Name: Aditi
Roll.No: 02
Div: A */

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();
                nr=sc.nextInt();
                need=new int[np][nr];
                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();
                        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();
                        System.out.println("Enter available matrix -->");
                    for(int j=0;j<nr;j++)
                        avail[0][j]=sc.nextInt();
                sc.close();
            }
            private int[][] calc_need()
            {
                for(int i=0;i<np;i++)
                    for(int j=0;j<nr;j++)
                        need[i][j]=max[i][j]-allocate[i][j];
                        return need;
            }
            private boolean check(int i)
            {
```


Bankers.java - Visual Studio Code [Administrator]
Bankers.java
C: > Program Files > Java > jdk-10.0.1 > bin > Bankers.java
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Java

```java
                    for(int j=0;j<nr;j++)
                        need[i][j]=max[i][j]-allocate[i][j];
                        return need;
            }
            private boolean check(int i)
            {
                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)
                {
                    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(j==np)
                System.out.println("\nSafely allocated");
            else
                System.out.println("All proceess cant be allocated safely");
    }
    public static void main(String[] args)
     {
        new Bankers().isSafe();
    }
}
```



```
C:\Program Files\Java\jdk-10.0.1\bin>javac Bankers.java

C:\Program Files\Java\jdk-10.0.1\bin>java Bankers
Enter no. of processes and resources : 3
2
Enter allocation matrix -->
2 3 4
3 5 1
Enter max matrix -->
4 5 6
2 0 0
Enter available matrix -->
3 2 2
Allocated process : 0
Allocated process : 1
Allocated process : 2

Safely allocated

C:\Program Files\Java\jdk-10.0.1\bin>
```