PROGRAMMING USING JAVA
WEEK 10 ASSIGNMENT

1.Write a Java program for the tasks printA, printB, and print100 are executed simultaneously to display the letter a 100 times, the letter b 100 times, and the numbers from 1 to 100.

```
import java.io.*;
import java.util.*; import
java.text.*; import
java.math.*; import
java.util.regex.*;

class GFG {
        // Prints numbers from 1 to n
        static void printNos(int n)
        {
                if(n > 0)
                {
                        printNos(n - 1);
                        System.out.print(n + " "); }
                return;
        }

        // Driver Code
        public static void main(String[] args) {
                printNos(100);
        }
```

} **Output :**

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

2.Demonstrate the concept of Threads to achieve multitasking program using Thread class and Runnable Interface (separately) for the below scenario:
Create an Array of 9 numbers. And create three Threads to split the task evenly among the three threads. And each thread has to add up and report the answer to the main thread where the main thread waits for the 3 threads and computes the summation of all the three threads. Note: Assign names to the threads as well.

```
class MyThread implements Runnable { String
name;
```

```
Thread t;
    MyThread (String thread){

    name = threadname;
    t = new Thread(this, name);
System.out.println("New thread: " + t);
t.start();
}


public void run() {
 try {
     for(int i = 5; i > 0; i--) {
     System.out.println(name + ": " + i);
      Thread.sleep(1000);
}
}catch (InterruptedException e) {
     System.out.println(name + "Interrupted");
}
     System.out.println(name + " exiting.");
} }

class MultiThread {
public static void main(String args[]) {
     new MyThread("One");
     new MyThread("Two");
     new NewThread("Three");
try {
     Thread.sleep(10000);
} catch (InterruptedException e) {
      System.out.println("Main thread Interrupted"); }
      System.out.println("Main thread exiting."); }
```

} **Output :**

```
New thread: Thread[One,5,main]
 New thread: Thread[Two,5,main] New
 thread: Thread[Three,5,main] One: 5
 Two: 5
 Three: 5
 One: 4
 Two: 4
 Three: 4
 One: 3
 Three: 3
 Two: 3
 One: 2
 Three: 2
```

Two: 2
One: 1

Three: 1
Two: 1
One exiting.
Two exiting.
Three exiting.
Main thread exiting.