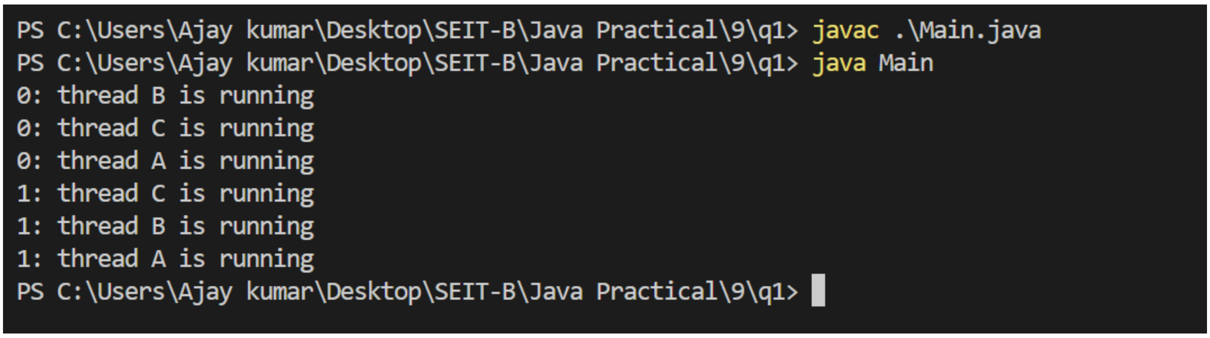
1.Write a Java program to implement multithreading using any three classes that will run concurrenly.

Code:

```
class A extends Thread {
  public void run() {
    for (int i = 0; i < 2; i++) {
      System.out.println(i + ": thread A is running");
    }
  }
}
class B extends Thread {
  public void run() {
    for (int i = 0; i < 2; i++) {
      System.out.println(i + ": thread B is running");
    }
  }
}
class C extends Thread {
  public void run() {
    for (int i = 0; i < 2; i++) {
      System.out.println(i + ": thread C is running");
    }
  }
}
class Main {
  public static void main(String args[]) {
    A a = new A();
    B b = new B();
    C c = new C();
    a.start();
    b.start();
    c.start();
  }
}
```

Output:



```
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\q1> javac .\Main.java
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\q1> java Main
0: thread B is running
0: thread C is running
0: thread A is running
1: thread C is running
1: thread B is running
1: thread A is running
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\q1>
```

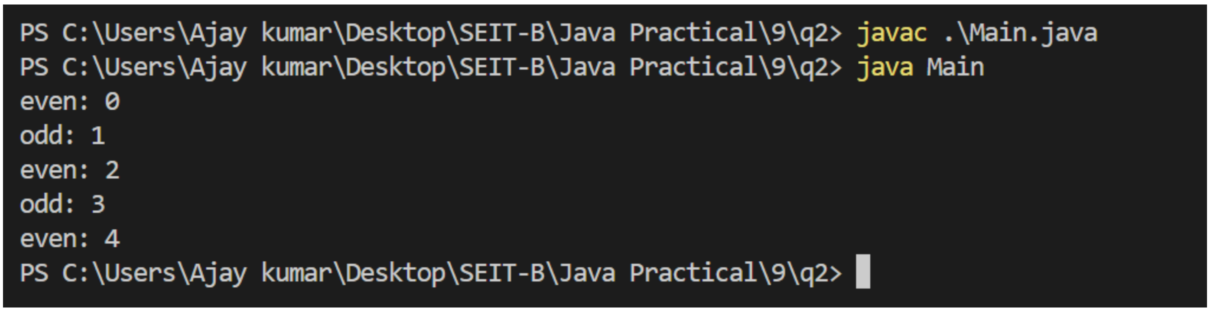
2. Create two threads such that one thread will print even number and another will print odd number in an ordered fashion.(Use Thread Class)

Code:

```
class odd extends Thread {
  public void run() {
    for (int i = 1; i < 5; i += 2) {
      System.out.println("odd: " + i);
    }
  }
}
class even extends Thread {
  public void run() {
    for (int i = 0; i < 5; i += 2) {
      System.out.println("even: " + i);
    }
  }
}
class Main {
  public static void main(String args[]) {
    odd a = new odd();
    even b = new even();

    a.start();
    b.start();
  }
}
```

Output:



```
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\q2> javac .\Main.java
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\q2> java Main
even: 0
odd: 1
even: 2
odd: 3
even: 4
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\q2> 
```

1. Write java program to print Table of Five, Seven and Thirteen using Multithreading(Use Runnable Interface)

Code:

```
class Five implements Runnable {
  public void run() {
    for (int i = 1, j = 5; i <= 10; i++, j += 5) {
      System.out.println("5 x " + i + " = " + j);
    }
  }
}

class Seven implements Runnable {
  public void run() {
    for (int i = 1, j = 7; i <= 10; i++, j += 7) {
      System.out.println("7 x " + i + " = " + j);
    }
  }
}

class Thirteen implements Runnable {
  public void run() {
    for (int i = 1, j = 13; i <= 10; i++, j += 13) {
      System.out.println("13 x " + i + " = " + j);
    }
  }
}

class Main {
  public static void main(String args[]) {
    Five five = new Five();
    Seven seven = new Seven();
    Thirteen thirteen = new Thirteen();

    Thread t1 = new Thread(five);
    Thread t2 = new Thread(seven);
    Thread t3 = new Thread(thirteen);

    t1.run();
    t2.run();
    t3.run();

  }
}
```

Output:

```
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\PostQ1> javac Main.java
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\PostQ1> java Main
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
13 x 1 = 13
13 x 2 = 26
13 x 3 = 39
13 x 4 = 52
13 x 5 = 65
13 x 6 = 78
13 x 7 = 91
13 x 8 = 104
13 x 9 = 117
13 x 10 = 130
PS C:\Users\Ajay kumar\Desktop\SEIT-B\Java Practical\9\PostQ1>
```