CODING

Write a program to find the first 100 Prime numbers

```
import java.util.Scanner;

public class PrimeNumbers {
    public static boolean isPrime(int num) {
        for (int i = 2; i <= num / 2; i++) {
            if (num % i == 0)
                return false;
        }
        return true;
    }

    public static void main(String[] args) {
        int num;
        int count = 0;
        int startNumber = 1;

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        num = sc.nextInt();
        System.out.print("First " + num + " prime numbers are:\n");

        while (count < num) {
            startNumber++;
            if (PrimeNumbers.isPrime(startNumber)) {
                System.out.println(startNumber);
                count++;
            }
        }
    }
}
```

Now write all possible test cases to test your program. (Manual test cases: covering both positive and negative scenarios)

Test Cases:

- Test for int input where num > count for e.g. num = 1 or 100. For this positive test scenario the code prints first 100 prime numbers successfully.
- Test input for boundaries for values such as num = count for e.g. num = 0 and largest acceptable number for e.g. num = 11234567890. This code does not handle these values and returns no output for it.
- Test input for num < count for e.g. num = -1. The code returns no input since count has been initialized from 0.
- Testing the input for different data types. For e.g. num = 9.45 or three or “2” or any special character. This code only accepts integer values as input and does not return any output for float, string and other data types.