

**CSCE 212**  
**Project 1**  
**Simple Primality Test**  
**Due Date: 10/3**

**Introduction**

The objective of this lab is to write a short program in MIPS assembly and verify its correctness using simulation with MARS. This lab is designed to practice writing loop and if-statement instruction sequences.

**Design Requirements**

Write a program in MIPS assembly that (1) takes, as console input from the user, a signed 32-bit integer, (2) performs a test to determine if the number is prime, and (3) returns the result of the test to the user.

The program should not generate any type of prompt, and its output should be "prime" or "nonprime" followed by a newline character, denoting the result of the test.

The program must perform these steps in an infinite loop until the user terminates the program by providing an input value of zero.

Your program must use the algorithm described in the Implementation section.

**Implementation**

Use the pseudocode algorithm below to implement your primality test:

```
if (number < 2) prime=false;
else if (number == 2) or (number == 3) prime=true;
else if (number is even) prime=false;
else if (number mod 3 == 0) prime=false;
else {
    // find floor(sqrt(number))
    sqrt = -1;
    val = 1;
    sum = 0;
    while (sum <= n) {
        sum = sum + val;
        val = val + 2;
        sqrt = sqrt + 1;
    }

    // check for divisibility by (6k +/- 1) for all k
    prime=true;
    val = 5;
    while (val <= sqrt) {
        if (number mod val == 0) or (number mod val+2 == 0) {
            prime=false;
            break;
        }
        val = val + 6;
    }
}
```

**What to Submit**

Submit your code via CSE Dropbox (via <https://dropbox.cse.sc.edu>)