

Introduction

This assignment is about assembly language programming using the MIPS simulator called SPIM. To learn how to use the simulator, watch the relevant video in the video series for the course. To learn the basics of assembly language programming, read the appendix from Patterson+Hennessy and try out the sample programs posted to resources.

Question 1

Write a SPIM program to determine if an integer is a palindrome or not.

A palindrome number is a number that reads the same from the front and the back. Examples are: 212, 44, 9009, 4567654. To calculate whether a number is a palindrome or not, you can first reverse the number (using repeated division with a **div** opcode and a loop) and then check for equality.

Sample I/O:

```
Enter a number:
200
It is not a palindrome
```

Sample I/O:

```
Enter a number:
373
It is a palindrome
```

Save your program as **question1.asm**

Question 2

Write a SPIM program to determine if an integer is prime or not.

A prime number is one that is only divisible by 1 and itself. Examples are: 3, 11, 313.

Sample I/O:

```
Enter a number:
200
It is not prime
```

Sample I/O:

```
Enter a number:
373
```

It is prime

Save your program as **question2.asm**

Question 3

Write a SPIM program to find all palindromic primes between two integers supplied as input (start and end points are excluded).

Some examples of palindromic primes are: 11, 191, 313

Sample I/O:

Enter the starting point N:

200

Enter the ending point M:

800

The palindromic primes are:

313

353

373

383

727

757

787

797

Save your program as **question3.asm**

Submit ALL asm files in a single ZIP file to the Automatic Marker.

Mark Weighting

- Question 1: 30
- Question 2: 30
- Question 3: 40