C335 Mini Programming Project

Points:	: 50 points
Due Date:	: March 10th
Submissions:	: For F-2-F students, hardcopy and c-copy to Canvas For online students, e-copy to Canvas

WHAT TO HAND IN:

- Your **source code** for each part.
    - ***You need to nicely document your code! Otherwise, you will lose points.***
- The **outpu**t of your program. You could *snap-shot* your screen, and paste it to a word document.
- Submit **E-copy** to Canvas (through "Assignments" Function).
- Face-2-Face students should also submit a hardcopy.

PART I (20 POINTS)

Using SPIM, write and test an adding machine program that repeatedly reads in integers and adds them into a running sum. The program should stop when it gets an input that is 0, printing out the sum at that point. Use the SPIM system calls covered in lecture notes (also on Text Appendix A.43 - A.45).

PART II (20 POINTS)

Write and test a MIPS assembly language program to compute and print the first 100 prime numbers. A number n is prime if no numbers except 1 and n divide it evenly. You should implement two routines:

- **test_prime(n),** return 1 if n is prime and 0 if n is not prime
- **main(),** iterate over the integers, testing if each is prime. Print the first 100 prime numbers

You might want to write a C program for solving the same problem first, and then convert it to MIP assembly. Test your programs by running them on SPIM.

Hint: here is the example code for % operation:

```
div $a0, $t0
mfhi $t1          #$t1 = $a0 % $t0
```

PART III (10 POINTS)

Using SPIM, write and test a recursive program for solving the classic mathematical recreation, the Towers of Hanoi puzzle. The puzzle consists of three pegs (1, 2, and 3) and n disks (the number n can vary; typical values might be in the range from 1 to 8). Disk 1 is smaller than disk 2, which is in turn smaller than disk 3, and so forth, with disk n being the largest. Initially, all the disks are on peg 1, starting with disk n on the bottom, disk n – 1 on top of that, and so forth, up to disk 1 on the top. The goal is to move all the disks to peg 2. You may only move one disk at a time, that is, the top disk from any of the three pegs onto the top of either of the other two pegs. Moreover, there is a constraint: You must not place a larger disk on top of a smaller disk.

The C program below can be used to help write your assembly language program.

```
/* move n smallest disks from start to finish using extra */
void hanoi(int n, int start, int finish, int extra)
{
        if(n != 0){
                hanoi(n-1, start, extra, finish);
                print_string(“Move disk”);
                print_int(n);
                print_string(“from peg”);
                print_int(start);
                print_string(“to peg”);
                print_int(finish);
                print_string(“.\n”);
                hanoi(n-1, extra, finish, start);
        }
}

main()
{
        int n;
        print_string(“Enter number of disks>“);
        n = read_int();
        hanoi(n, 1, 2, 3);
        return 0;
}
```

Work on your own! You might be asked to explain your code to the Instructor.