CS220
Introduction to Computer Organisation
Lab 5

Amey Karkare

We take a detour from the design of SMIPS processor and work on assembly programming for next 2 labs. You will be implementing MIPS assembly programs for given problems and run them on SPIM simulator[1]. Refer to MIPS manual for instruction set, and SPIM manual for running the programs.

The following sections describes some interesting functions/conjectures that you have to implement in MIPS assembly programs.

NOTE:

1. For each problem, **20%** marks are reserved for **useful** comments. Comments like

   ```
   add  $11, $12, $13   # adding $12 to $13, storing in $11
   ```

   are useless, and should be avoided.

2. Usage conventions for MIPS registers have to be honored.

3. For each task, the function to be implemented has to be free of input/output calls. Parameter should be passed as argument, and result should be returned as return value. Error reporting using output call is ok, but should not clutter your main logic.

4. For each task, you have to additionally implement a `main` function that reads input and produces output. This is *not* mentioned in the individual tasks, but you have to do it.

5. You are encouraged to implement and use helper functions to create a modular design.

6. Assume the inputs to be non-negative (≥ 0) for each task, so there is no need to test for negative numbers.

[1] Since SPIM simulator supports almost all of the MIPS instruction set, the programs you write need not be restricted to SMIPS instruction set.

1 Euler's *totient* function $\phi(m)$ [10]

Euler's totient function $\phi(m)$ is defined as the number of positive integers r $(1 \leq r < m)$ that are *coprime* to m. Two numbers are *coprime* if their greatest common divisor equals 1.

$\phi(1)$ **is defined to be** 1 **as a special case**.

Example: For $m = 10$: $r = 1, 3, 7, 9$; thus $\phi(m) = 4$.

You have to implement a function named `totient` in the file `totient.asm`.

2 GoldBach's Conjecture [10]

Goldbach's conjecture is one of the oldest and best-known problems in number theory and in all of mathematics. It states:

Every even integer greater than 2 *can be expressed as the sum of two primes.*

You have to write a function `goldbach` in file `goldbach.asm` that takes an input n, and returns two prime numbers[2] p_1 and p_2 that sum up to n, if n is positive even integer, otherwise it returns -1 and -1.

3 McCarthy's 91 function [10]

Implement function `mccarthy91` in the file `mccarthy.asm`. The McCarthy 91 function is defined as:

$$mccarthy91(n) \quad = \quad \begin{cases} n - 10 & \text{if } n > 100 \\ mccarthy91(mccarthy91(n + 11)) & \text{if } n \leq 100 \end{cases}$$

[2] Remember you can use registers \$v0 and \$v1 to hold *two* return values.