CSE 341 Project

Milestone Due on May 1, 2018
Final Due on May 11, 2018

1. Project Description

You are required to write simple programs in assembly language in this project.

1). Calculator

Requirement You are asked to implement a simple calculator including addition, subtraction, multiplication and division.

2). Hamming Distance

Background Hamming distance measures the minimum number of substitutions required to change one string into the other, or the minimum number of errors that could have transformed one string into the other. For example, the distance between 1011 and 1001 is one, and the distance between 2345 and 2333 is two.

Requirement You are asked to implement code to calculate Hamming distance between two strings of numbers with length of two (e.g. 11 and 12).

3). Euclidean Distance

Background Euclidean distance defines the distance between two points in Euclidean space. When considering a one dimension Euclidean space, the distance between x and y could be given by:

$$\sqrt{(x-y)^2} = |x-y|.$$

For a two dimension Euclidean space, if

$$\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2)$$

The distance is given by:

$$\mathrm{d}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathrm{x}_1 - \mathrm{y}_1)^2 + (\mathrm{x}_2 - \mathrm{y}_2)^2}$$

Requirement You are asked to calculate Euclidean distance in a two dimension Euclidean space. For convenience, you should assume radicand is the square of an integer.

2. Submission Approach & Grading Criteria

Submission Approach

Submit your source file via UBlearns with name:

- *Calculator_firstname_lastname_UBID.asm*
- *Hamming_firstname_lastname_UBID.asm*
- *Euclidean_firstname_lastname_UBID.asm*
- *Report_firstname_lastname_UBID.pdf*

Grading Criteria

- Calculator (addition, subtraction, multiplication, division): 10 % for each
  (Bonus: +10 points if submitted before Milestone Due)
- Hamming distance: 20 %
- Report: 20 %
- Euclidean distance: **20 %**
  (Bonus: + 5 points if you implement without using sqrt)

3. Skeleton Code

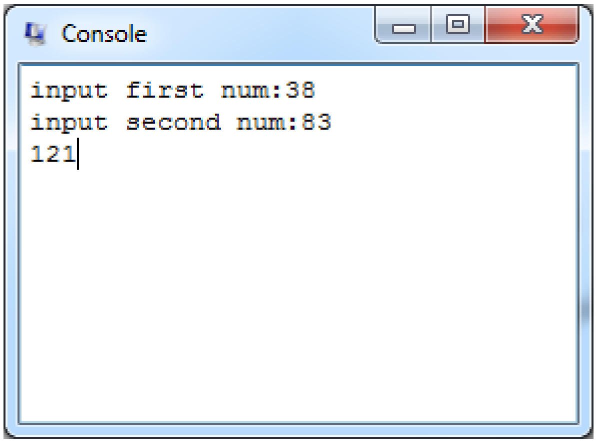
You need to use the console to input number and also output your result to the console, otherwise you cannot be graded. For example, this is the format of calculator:

```
input first num: 30
input second num: 15
input operation (1∼4,1:+,2:-,3:×,4:÷): 3

result is: 450
```

We will give the skeleton code of addition:

skeleton code of addition

```
          .data
first:    .asciiz  ”input␣first␣num:”
second:   .asciiz  ”input␣second␣num:”
          .text
main:
          li       $v0,    4
          la       $a0,    first
          syscall

          li       $v0,    5
          syscall
          move     $s0,    $v0

          li       $v0,    4
          la       $a0,    second
          syscall

          li       $v0,    5
          syscall
          move     $s1,    $v0

          add      $a0,    $s0,    $s1
          li       $v0,    1
          syscall

          li       $v0,    10
          syscall
```



```
input first num:38
input second num:83
121
```