University at Buffalo

Project 3

Due date: Friday, March 28, 2025 at 11:59 PM

Note:

To write assembly program you can use any text editor such as Notepad. At this point, the structure of the assembly program should be similar to the structure below. You can add your code as mentioned below inside the structure. Also, in assembly program, any line that starts with # will be considered as a comment and the assembler will not even execute it.

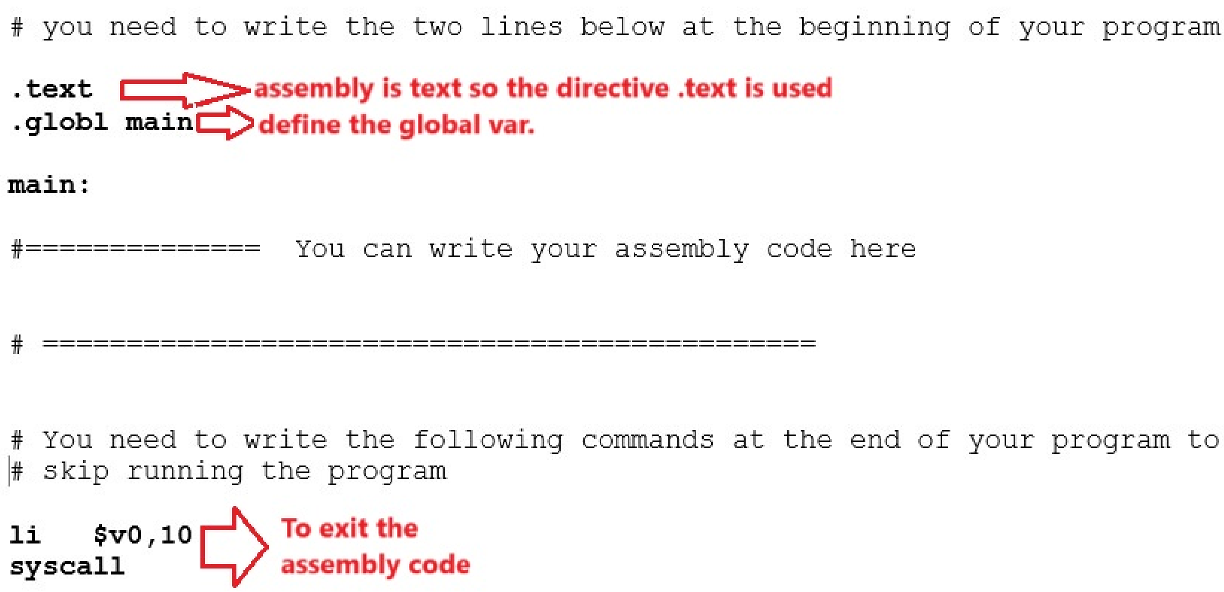

Figure 1: Simple structur of the assembly program.

Question 1

[15 Points] Write MIPS assembly code that is used to convert centimeter (cm) to meter (m). As you know the 1 $m = 100\ cm$, therefore, to convert form centimeter to meter we need to divide the centimeter value by 100 as shown below:

$$m = \frac{cm}{100} \tag{1}$$

Write MIPS assembly code to convert the length which is given in cm to meter. Note that the length can be fractional number, and it is not always an integer. Therefore, use IEEE754 single precision standard to define all the numerical values in this problem.

You will need to perform the following steps:

1. Convert decimal 100.0 to its equivalent single precision binary representation and store the result into any integer (INT) register, for example, $t0. Note that the value inside $t0 now is not the real number (100.0) because $t0 is not used with fractional numbers, but we will use it to store the binary of the real number 100.0 (defined using single precision) as an integer number.
2. Store $t0 in the memory location starts at 0x10010000
3. Load the binary of 100.0 from the memory (which is stored in the previous step) into any FP register, say $f0 using "lwc1" instruction.
4. Now, convert 350.0 cm to meter to test the system. First we need to store the 350.0 cm into any FP register, then in the second stage, we can perform the division. To do this follow the steps below:
   (a) Convert 350.0 to its single precision binary equivalent and store it in any INT register as a binary integer number, use register $t1 here.
   (b) Now store $t1 (i.e 350.0) into the memory, for example use memory location starts at 0x10010100.
   (c) Use "lwc1" instruction to load 350.0 from memory into any FP register, use register $f1 here.
   (d) Now use "div.s" instruction to perform the division ($f1/$f0) and store the result in $f2 register.

Simulate your code using QtSpim to assure that it works properly. Name your program Prj3_Q1.s and submit .s file via UBlearn.

Question 2

[15 points] A second-order (quadratic) equation is a polynomial equation of the form:

$$Ax^2 + Bx + C = 0 \tag{2}$$

where:
- A, B, and C are constants,
- $A \neq 0$,
- x is the unknown variable.

You can the roots of the second order equation using the quadratic formula below:

$$x = \frac{-b \pm \sqrt{B^2 - 4AC}}{2A} \tag{3}$$

where:
$B^2 - 4AC$ is called the discriminant (Δ). The nature of the roots depends on the discriminant:

If $\Delta > 0$, there are two distinct real roots.
If $\Delta = 0$, there is one real repeated root.
If $\Delta < 0$, there are two complex roots.

We will assume that the discriminator $\Delta \geq 0$ in our problem so there are one or two real root(s) for the quadratic equation.

Write MIPS assembly code to find the real roots of the quadratic equation below:

$$x^2 - 3x + 2 = 0 \tag{4}$$

To start the solution, store A, B, and C in registers $f0, $f1, and $f2, respectively. Then solve the equation and store the first root of the quadratic equation in register $f5 and in the memory starting from address 0x10010100 and the second root in register $f4 and memory starting from address 0x10010200.
Use IEEE 754 single precision format to define all the real numbers in this problem.
Assume that the result of any calculation in the solution will not be more than 4 bytes.
Simulate your code using QtSpim to find the roots of the equation. Check if you got the right roots. Name your program Prj3_Q2.s and submit .s file via UBlearn.