**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**
**UNIVERSITY OF BRITISH COLUMBIA**
**CPEN 211 Introduction to Microcomputers, Fall 2022**
**Lab Proficiency Test 2 (LPT-2) – [5 Marks]**
***November 25th, 2022: 4:10pm-5:40pm. Code to be submitted via Canvas at 5:40pm***

## 1 Overview

Your LPT-2 is available on Canvas. It has three files.

- This pdf file.
- The Arm v7 instruction set as a separate pdf (for your reference).
- The q1.s file. You will edit this file.

You may use simulator listed in the following link to compile and evaluate your code.
**Simulator Link**: https://cpulator.01xz.net/?sys=arm-de1soc&d_audio=48000

## 2 Problem Statement

The function in Figure 2 computes the Highest Common Factor (HCF) of r0 and r1. This function in written in C. You will need to write the ARM assembly that implements this function.

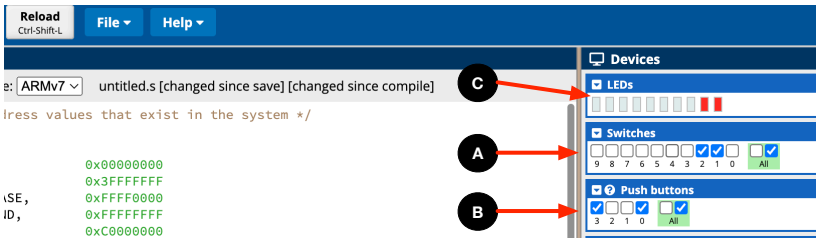
```
hcf(int r0, int r1) {
    if (r0 == 0 || r1 == 0){
        return;
    }
    else if (r0 == r1) {
        return;
    }
    else if (r0 > r1) {
        hcf(r0-r1, r1);
    }
    else {
        hcf(r0, r1-r0);
    }
    return;
}
```

Figure 1: A function written in 'C' to compute Highest Common Factor (HCF).

## 3 Inputs and Outputs

The 'hcf' function in Arm assembly uses registers r0 and r1 as its first and second parameter respectively.

- The register r0 contains the first parameter. Its initial value is the value in switches SW0-SW9 (A). You can vary the values of SW0-SW9 to test your code.
- The register r1 contains the second parameter. Its initial value is the value in push buttons KEY0-KEY3 (B). You can vary the values of KEY0-KEY3 to test your code.
- For this LPT, assume that registers r0 and r1 contain only non-negative initial values.
- Except in cases when either r0 or r1 have an initial value of binary '0', both r0 and r1 should hold the final hcf value upon returning to the main calling code.
- The final hcf is displayed on the LEDs (C).




## 4 Arm Assembly Implementation: q1.s

- Edit the q1.s file in this LPT. Do not create your own q1.s file. You will get ZERO marks if you do.
- This portion in Figure 2 describes where you will write your code. This is at the end of the q1.s file.

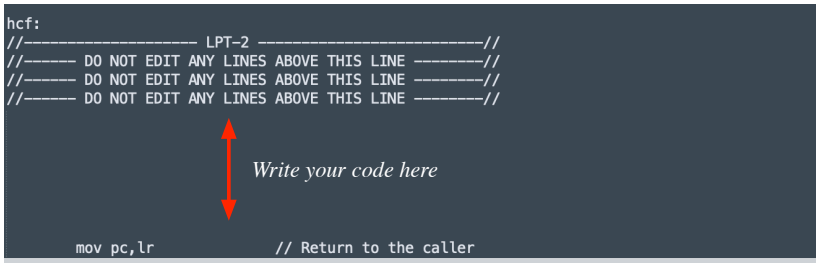



Figure 2: The 'unfilled' hcf function in q1.s. Fill only the portion indicated in this figure. This part is at the end of the file. Do not edit other parts.

- Do not edit other parts of q1.s. You will get ZERO marks if you do.
- Do not edit the LABEL `hcf:`. You will get ZERO marks if you do.

## 5 Important Points

- Your file **must** be named as q1.s when you submit to canvas. If canvas renames this file it is OK.
- You must submit the entire q1.s file. Do not submit a file with ONLY the `hcf` function and nothing else.
- You are strongly encouraged to test your code. You can do this by varying SW0-SW9 and thereby varying the values in r0. You can vary KEY0-KEY3 and thereby vary the values in r1.
- The final 'hcf' values will be displayed on LED0-LED9 on the right hand side of your screen.
- Upload your q1.s file by attaching it as your solution to Question under "Lab Proficiency Test 2" on Canvas before 5:40 pm.
- Submit early and resubmit as needed. No submissions will be accepted after 5:40 pm (emailed submissions will not be accepted).
- We will not manually grade you.
- Do not create your own q1.s file. Use the q1.s file that you downloaded from Canvas. If you create your own q1.s file, you will get ZERO points.

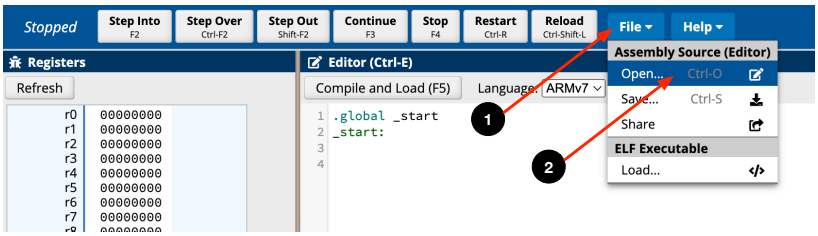
## 6 How to not LOSE POINTS

You will lose points for each of the following conditions – the minimum points you can get is ZERO and the maximum is FIVE:

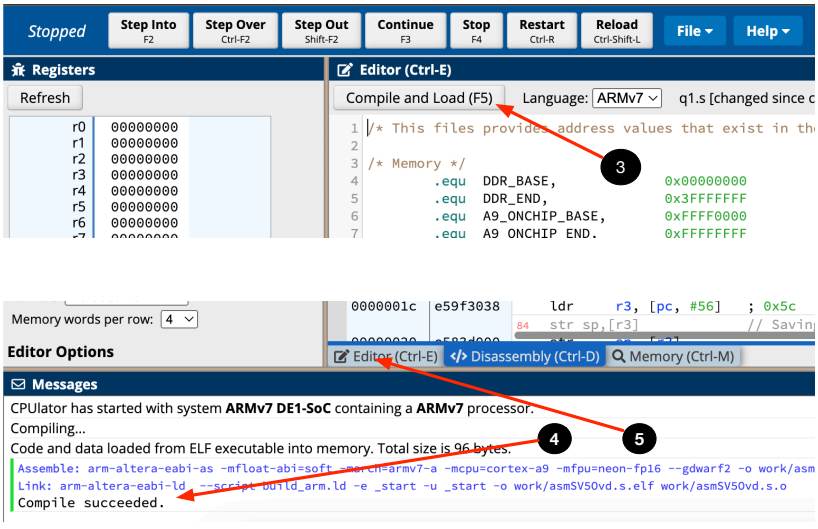
- You did not submit a file named q1.s on canvas. **You will get a ZERO if this happens**.
- You did not submit the entire q1.s file (e.g. your q1.s file has ONLY the hcf function and not the text before it, etc.). **You will get a ZERO if this happens**.
- Your q1.s file does not compile (e.g., due to syntax errors). **You will get a ZERO if this happens**.
- Your function is stuck in an infinite loop and does not return. **You will get a ZERO if this happens**.
- You edited the q1.s file in the parts above the `hcf:` function or changed the LABEL `hcf:`. This is not allowed and **you will get a ZERO if this happens**.
- Your 'hcf' implementation in q1.s produces incorrect outcomes for the different values in r0 and r1 in our autograder. **You will lose 1 point for each pair of inputs that produce incorrect outcomes**.

## 7 Using the Online Arm Simulator

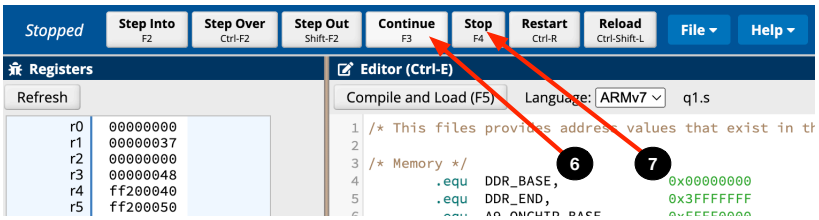
**1. Load q1.s**: Kindly edit q1.s on your local computer. Once you have finished editing, you can go to the simulator website. You can click `File` ❶ and `Open` ❷ to select q1.s.




**2. Compile**: Compile your code by clicking the `Compile and Load` ❸ button at the top of the page. Check if your compilation was successful ❹ at the bottom of the page. You can then go to the editor ❺.



**3. Execute**: Execute your code by clicking the `Continue` 6 button at the top of the page. Stop your execution using the `Stop` 7 button at the top of the page.




**4. Save**: Save your file by clicking the `Save` 8 button within the `File` button.


## 8 Appendix: Example HCF Calculation

The highest common factor (HCF) of two numbers is the largest number which is a factor of both those numbers.

**Example**
What is the HCF of 28 and 21?
**Answer**

- List down the factors of 28. The number 28 can be written as the product of the following numbers.

$$28 = 1 \times 28 \tag{1}$$
$$28 = 2 \times 14 \tag{2}$$
$$28 = 4 \times 7 \tag{3}$$

  Thus, the factors of 28 are 1, 2, 4, 7, 14, and 28.
- Next, list down the factors of 21. The number 21 can be written as the product of the following numbers.

$$21 = 1 \times 21 \tag{4}$$
$$21 = 3 \times 7 \tag{5}$$

  Thus, the factors of 21 are 1, 3, 7, and 21.
- By comparing the factors of 28 and 21, we find that their common factors are 1 and 7.
- Between these common factors, observe that 7 is greater than 1.
- Therefore, 7 is the highest common factor (HCF) between 28 and 21.