

LAB C

Control Flow Operations

The purpose of this lab is to learn basic register-register type, immediate and branch type instructions. Please write the assembly program for each question and take screenshots of each stage of your program to show that the processes were performed successfully. Demo your work to TA for your grade to be recorded and upload your report (including screenshots and your discussion) to moodle.

Q1: You wish to calculate the total marks for a course that has the following portions:

- Assignment 1 – 5%
- Assignment 2 – 5%
- Assignment 3 – 10%
- Mid-term exam – 20%
- Project – 20%
- Final exam – 40%

Students who don't submit Assignment 1 will have their marks for the Assignment 2 increased to 10%. Students who don't submit Assignment 2 will have their marks for the Assignment 1 increased to 10%. Students who don't attend Mid-term exam will have a final exam worth 60%

Write assembly code that loads up the full marks for each portion described above into registers starting from x5 (x5, x6. ...). Design if-else blocks that allow calculation of marks for students who miss assessment as mentioned above. Optimize your design to use less registers (but do not over-write the registers that hold values for each portion) or to have less instructions. Discuss how you optimized your design. You are assuming that the student is getting full marks for each portion. Calculate and save the final result in x11. Show your design and simulation to the Teaching Assistant.

Q2: Write assembly code that calculates the triangular number for a given number that is stored in x20. Use equality branching instructions to perform iterative additions to come up with the final sum which is the triangular number.

Example: if x20 =5, triangular number = $5 + 4 + 3 + 2 + 1 = 15$

if x20 =8, triangular number = $8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 36$

Appendix:

Q1:

The first section of your code loads the marks for each portion of the course marks into registers starting from x5, x6, ...

In the second portion of your code, you should create branch instructions that checks if x5 is zero or if x6 is zero. If either of them is zero, one of the registers get assigned value of 10. It doesn't matter which one of the registers is assigned the value of 10 as long as the other is 0.

In the third portion of your code, you create a branch instruction that checks if the register for midterm exam is zero. If it is, then you increase content of the final exam marks from 40 to 60.

In the fourth portion of your code, all the register contents get added together to produce the total for the course. Regardless of whether any branch instructions are implemented, this total must always come to 100.

Once your code is written, test it with the values of each course portion as shown in

- 1A) Show that the total in register x11 is 100. Screen capture and copy into report. (2 marks)
- 1B) Add instruction after the first section that change x5 to become zero so that the branch instruction is implemented. Assign register x2 to contain value '22' to show that the branch instruction was carried out. Screen capture and report. (2 marks)
- 1C) Add instruction to change x6 to become zero. Do the same as 1B. (2 marks)
- 1D) Comment out code written in 1B and 1C. Add instruction to make the register for mid-term exam become zero. Add value '33' to register x2. Screen capture. (2 marks)
- 1E) Combine 1D with 1B or 1C so that total value in x2 becomes '55'. Screen capture. (2 marks)
- 1F) Calculate alternative course work marks according to following method: mid-term exam = 20-5, project =20-5, final exam = 40+10. Update the appropriate

registers in this new code section, screen capture (make sure total is still 100%) and insert into your report. (2 marks)

Q2.

- 2A) Initialize one register with value of n, such as 5. Initialize the other register with value 0. This will be used to store the sum.
- 2B) Create the next portion of code which contains a looping branch. The branch condition should perform $\text{sum} = \text{sum} + n$ and $n-1$ as long as n is not zero. Screen capture and put in your report. (4 marks)
- 2C) Convert your code to calculate triangular numbers for negative numbers. (4 marks)