# Week 10 Lab A RISC-V Assembly language programming

## Objectives

Develop understanding and experience of:

1. Using branch instructions in RISC-V assembly language programming
2. Working on a programming challenge and comparing solutions in Java to assembly language

## Part A RISC-V Assembly language branches

What does the branch instruction **bne** mean?

Branch not equal

What does the Pseudo Instruction **beqz** mean?

Branch equal to zero

What does the branch instruction **b** do?

Branch

You now have **5 minutes** to type a short program in **Pseudocode** into this word document **(do not use RARS)** below.

The program should:

* Define a string.
* Print the string.
* Read a **Character** input from the user (not an integer input).
* Print out the character as its **integer** value.

NOTE: Characters are already stored as their ASCII integer value, so no conversions are needed. This is a recap of material from last week and the week before.

|  |  |  |
| --- | --- | --- |
| **Operation** | **Value needed in a7** | **Use of a0** |
| Print String | 4 | Address of string |
| Read Character | 12 | Character entered |
| Print Int | 1 | Int to print |

.data

String: text of string

.txt

Put the value 4 inside a7

Load address of string int a0

Ecall

Put value 12 in a7

Ecall

Put value 1 in a7

Ecall

*For the following questions, you might find it helpful to work in small groups and also consider how you would solve the problems in Java using loops and if/else. Please bear in mind that there isn’t a single correct answer to these exercises as there are multiple ways to approach them.*

1. Look at the countdown loop program that is given in the week 9 lecture (and in the concepts slides on branches). **You can find this in the Moodle area for last week in the PowerPoint of the lecture slides on slide number 21.** That program is very limited as it uses register **a0** for the countdown and just prints the countdown on one line with no spaces between.
   * Using that program (along with any code from previous exercises or examples), create a countdown program that asks the user for an input number and then counts down from that number to 1, leaving a **space**, **new line**, **or other character** between each digit.
   * You will need to store the input number in a register, make sure that you understand why you can’t just use a0 for this. You should use the **bnez** instruction as in the original code and use subroutines if appropriate.

Test your program **twice** using **two different** numbers as an initial input.

Paste a screenshot of your code here:

Paste a screenshot of your Run I/O here for test 1:

Paste a screenshot of your Run I/O here for test 2:

## Extension Exercises

*The following exercises have been adapted from codingbat (*[*https://codingbat.com/java*](https://codingbat.com/java) *) that you may have already used to practise Java programming. These exercises require some problem solving in how you would use branch instructions to control the flow of the program.*

*If you have used the codingbat website, you will see that each puzzle has a set of tests to check the possible situations. Make sure you test the possibilities with your assembly language program.*

1. Create a program that takes in two integers from the user, and outputs “Makes 10” if either of them is 10 or the sum of the numbers is 10. Otherwise, the program should output “No 10 here”.
2. We are having a party with amounts of tea and candy. Ask the user to input the amount of tea and the amount of candy (as integers). Output the outcome of the party which will be “bad”, “good” or “great”. A party is good if both tea and candy are at least 5. However, if either tea or candy is at least double the amount of the other one, the party is great. However, in all cases, if either tea or candy is less than 5, the party is always bad.