RMIT UNIVERSITY

Programming Studio-2 COSC2804

Assignment 1

Assessment Type	Individual assignment. Submit in class via Canvas. The last submission prior to the assignment deadline will be graded. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements in class.
Marks	30

1 Overview & Instructions

READ CAREFULLY THE INSTRUCTIONS BELOW:

1. Please **arrive at the time outlined in the document for your group** to set up your programming environment. What you use in our usual class activities and assignment work will suffice.
2. Place your photo or student ID on the table for inspection by teaching staff.
3. **The programming challenge will start after a summary of these instructions (15 minutes at maximum).**
4. Please download the skeleton code in LC-3 .asm files from "*Assignment 1: Learning LC-3 Assembly with Minecraft (Part 1)*" Canvas page.
5. After downloading the files, move them to your working directory.
6. **Try to complete the questions in the programming challenge.**
7. You must not modify any of the labels provided to you in the skeleton code. Each question will warn you against any such modifications.
8. We recommend using **VS Code** as your editor to complete the questions.
9. Once completed, **add all your .asm files only to a .zip archive**.
10. **Submit your .zip archive** using the submission link, available at: "*Assignment 1: Learning LC-3 Assembly with Minecraft (Part 1)*" Canvas page. You are allowed to submit multiple times within the deadline.
11. You can leave the teaching space if you are satisfied with your submission. However, you must check with our teaching staff before doing so and avoid disrupting other students still focusing on the challenge.
12. **The programming challenge will last 2 hours.**

PLEASE NOTE:

- This is an open book assessment; you are welcome to refer to any teaching materials. It is advised that you use those provided in this course, as they will be the most helpful.
- All mobile/smartphones, tablets, laptops, or any other electronic device for communication or any other use beyond the completion of the programming challenge (such as the use of AI Assistance - ChatGPT) will be strictly forbidden.
- **The assignment must be completed individually.**

- You will not be allowed to communicate with anyone during the test, and RMIT Academic Integrity policies will be strictly enforced during the challenge, as highlighted at https://bit.ly/2OXHACN.

2 Preliminaries

The aim of this assignment is to hone your LC-3 programming skills, using the game *Minecraft* as a testbed. This section will get you started with creating an LC-3 development environment and understanding how to write LC-3 programs that interact with Minecraft.

Setting up the tools required for carrying out this assignment.

Setup instructions can be found in the course Canvas shell. See the module Getting Started with LC-3, which is linked on the front page.

Communicating with Minecraft via LC-3.

LC-3 is a very simple language that offers no native way of communicating with Minecraft. To get around this, we have provided a modified LC-3 virtual machine that contains additional TRAP routines for this purpose. The additional TRAPs are summarised in the table below.

Trap Vector	**Alias**	**Function**	**Description**
0x28	chat/CHAT	postToChat(R0)	Outputs a null terminating string starting at the address contained in R0 to the Minecraft chat.
0x29	getp/GETP	player.getTilePos() --> R0, R1, R2	Gets the position of the "tile" that the player is currently on. The x, y and z coordinates are output in registers R0, R1 and R2 respectively.
0x2A	setp/SETP	player.setTilePos(R0, R1, R2)	This function moves the player to the tile (x, y, z) = (R0, R1, R2).
0x2B	getb/GETB	getBlock(R0, R1, R2) --> R3	This function retrieves the ID of the block at tile (x, y, z) = (R0, R1, R2) and returns it to R3.
0x2C	setb/SETB	setBlock(R0, R1, R2, R3)	This function changes the ID of the block at tile (x, y, z) = (R0, R1, R2) to the value stored in R3.
0x2D	geth/GETH	getHeight(R0, R2) --> R1	This function calculates the y-position of the highest non-air block at (x, z) = (R0, R2) and returns the value to R1.
0x27	reg/REG	printRegisters()	Outputs the current register values to the console.

Notes:

- `TRAP 0x27 (printRegisters)` is provided for debugging purposes, since unlike the LC-3 web simulator shown in class, the virtual machine included with the starter code does not provide an easy way of inspecting the register values.
- Function arguments and return values are passed via the registers. For example, `TRAP 0x2D (getHeight)` assumes that the *x* and *z* arguments are passed via registers `R0` and `R2` respectively and outputs the return value to `R1`.
- You should edit and submit *.asm files as instructed below.

3 LC-3 Programming Challenges

The questions have been removed from this version. You will receive a printed version with three questions on the day of the in-class assessment.