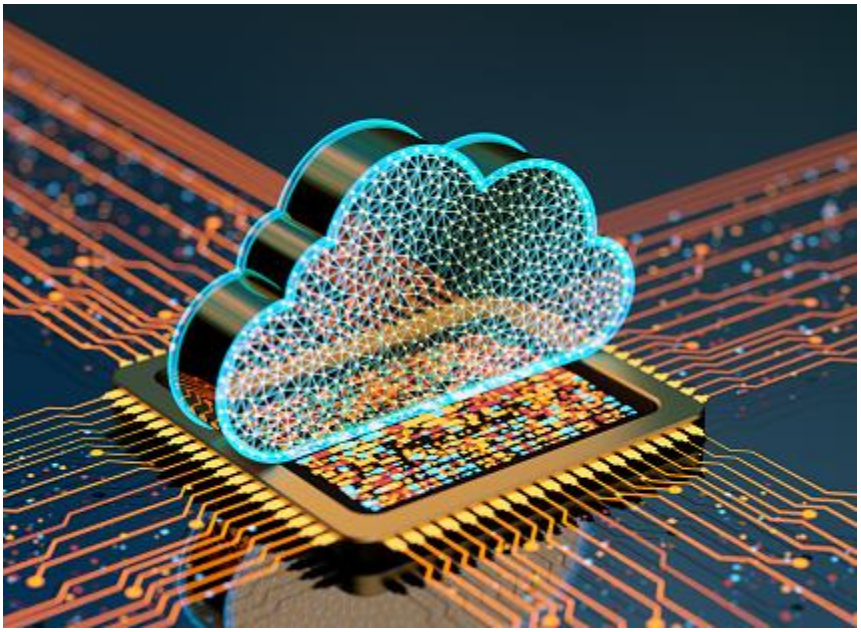


COS10004 – COMPUTER SYSTEM

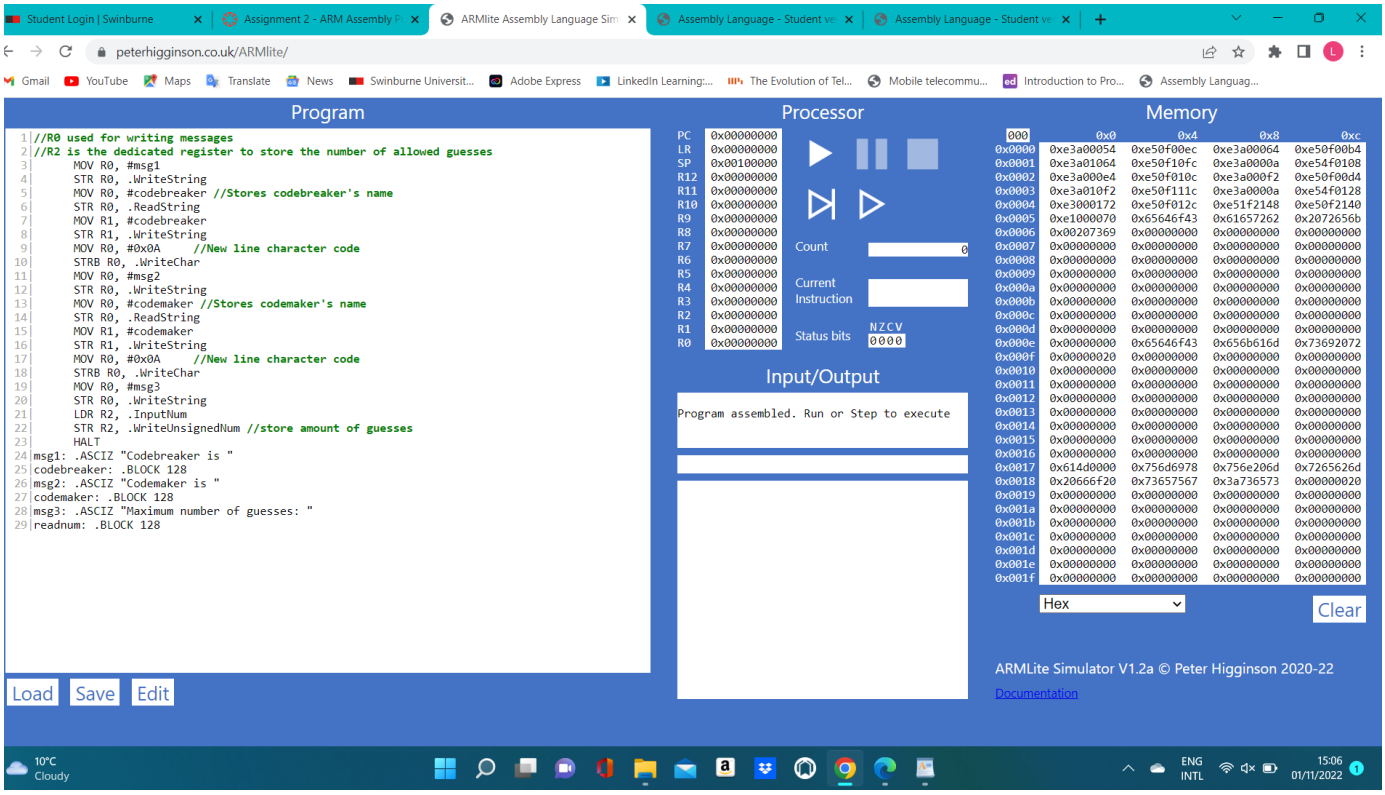
Lab Session: Monday 17:30 – 19:30 | Assignment 2 – Mastermind Game

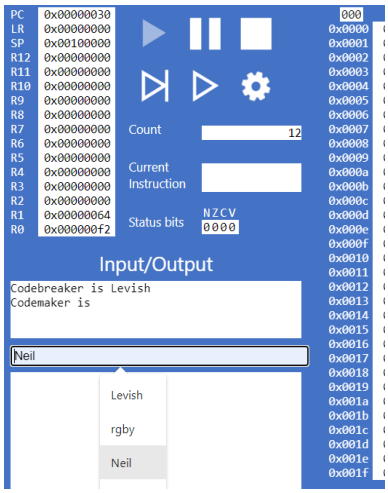
ABSTRACT

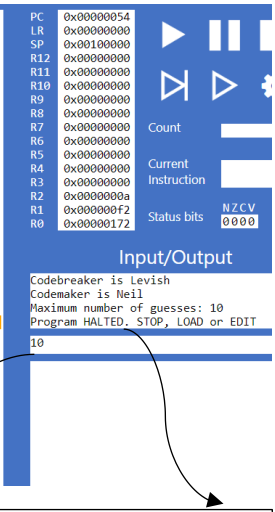
Levish Boodhoo
Student ID: 103818390

Mastermind – Stage 1

Game Setup







Flow of code asking for input and displaying outputs.

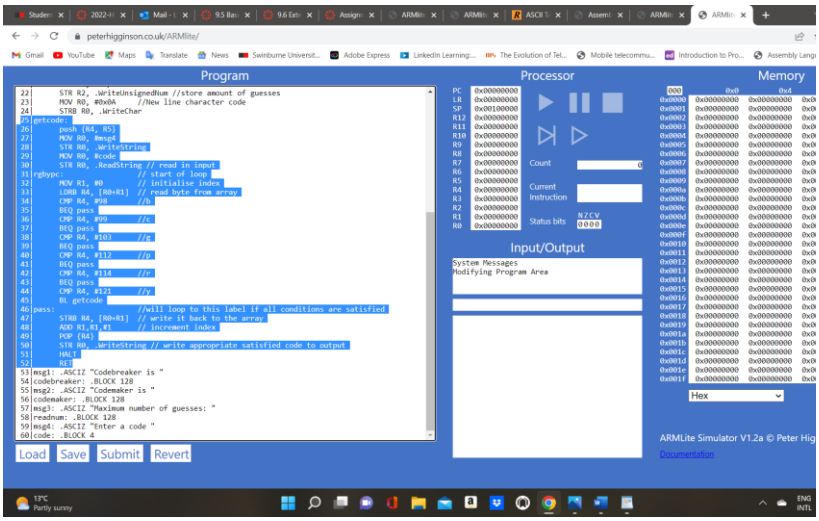
Input asking for the number of queries for future use.

Output lines asking for information to begin the game

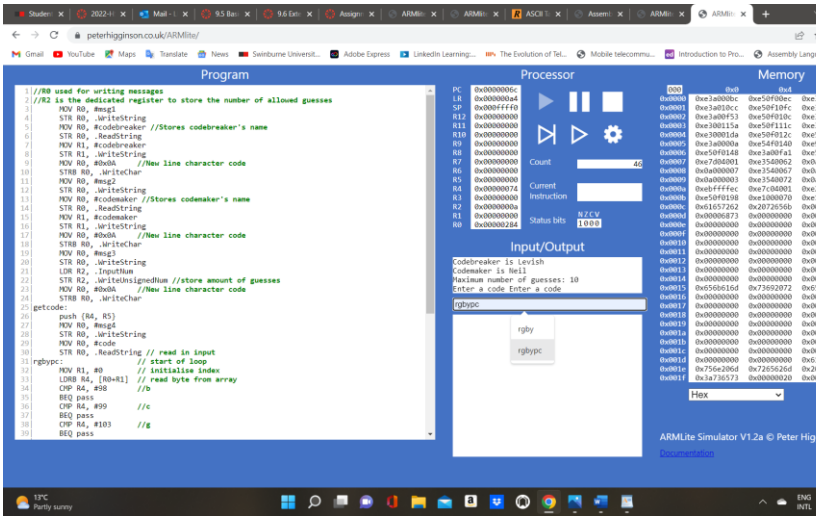
In this context, R0 is used to write messages or commands on the output display which asks for names and attempt numbers. However, there are labels used to store names entered by players such as #codebreaker and #codemaker. As far as query numbers are concerned, I have used a dedicated register to store the number (R2) which will be used only for the query number throughout the whole assembly code. As the code is processed stepwise, it will ask for inputs and as we press Enter, the line "Codebreaker is <name>" and other following lines will be displayed.

Mastermind – Stage 2

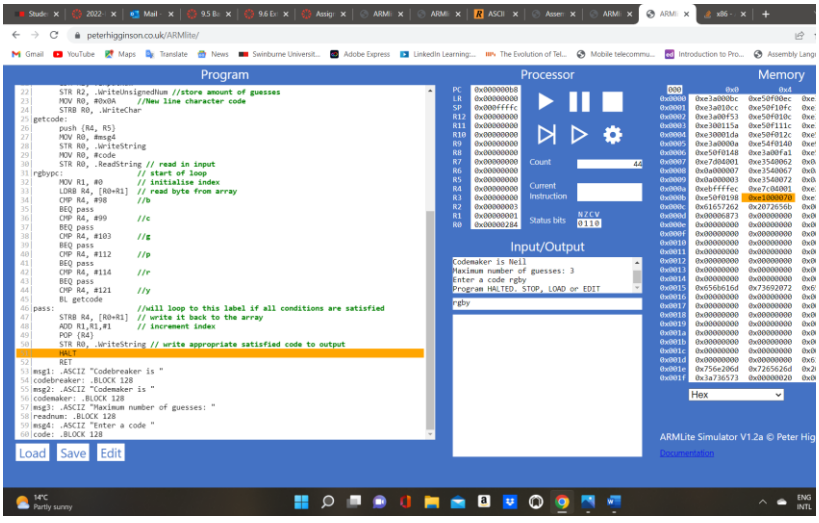
A code entry Function



This is the code for Stage2. There has been a functionality that I have faced problems creating which is limiting the code count to 4. However, I was able to define all the required code to allow the player to input code that consists only of "rgbypc".



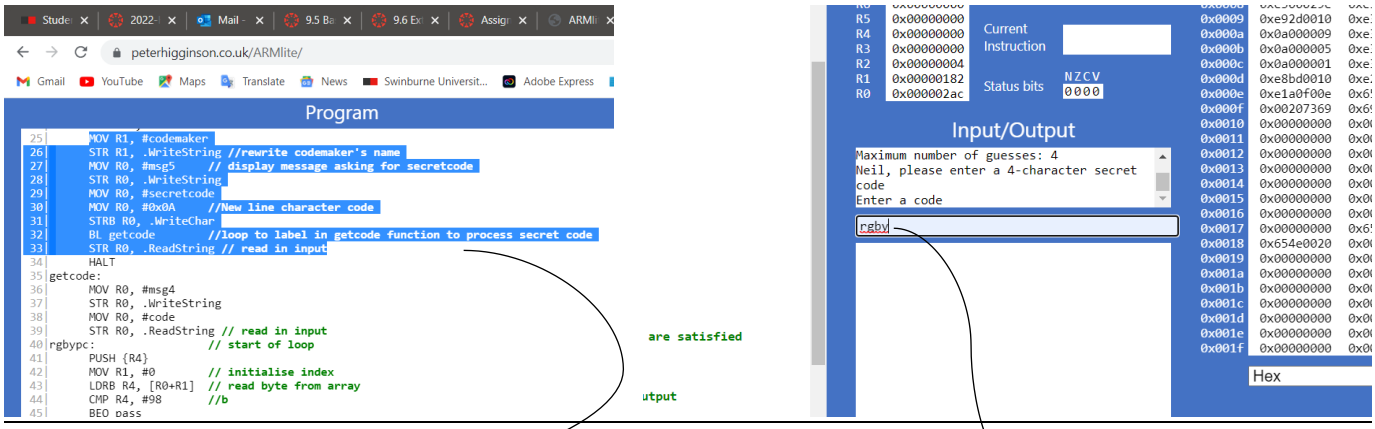
In this figure, I have tried to input alphabet other than "rgbypc" and as soon as the code get a code which does not satisfies the mentioned conditions, it will loop back to the start of the function, "getcode". There are other labels like "pass and" rgbypc"



This is the whole code functioning well. Only the 4-length code has been a limitation to me. Otherwise, the whole other code satisfies the required concept. It reads in the required lowercase alphabet code to display it after the line.

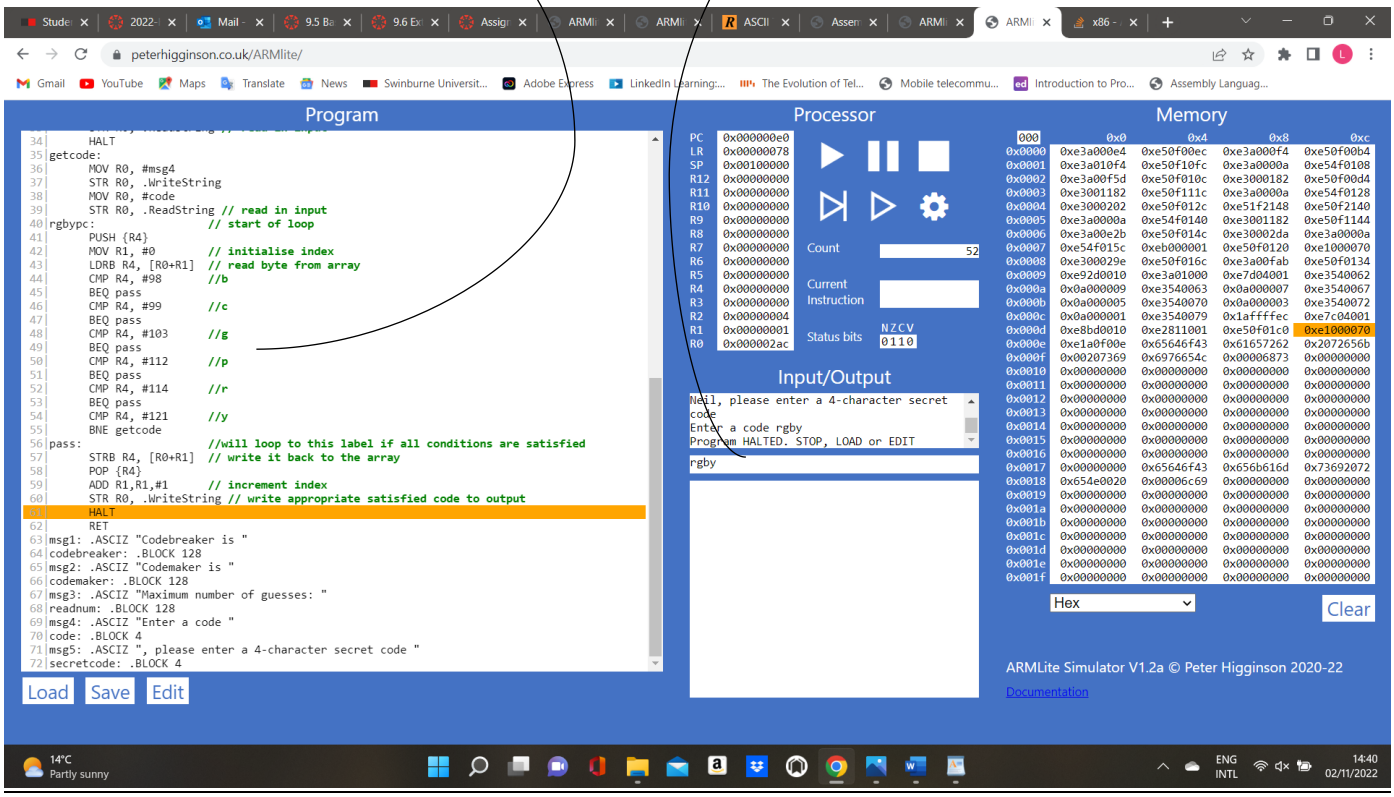
Mastermind – Stage 3

Getting the Secret Code



The displayed code asks the codemaker for the secret code and storing the information in an array called, #secretcode.

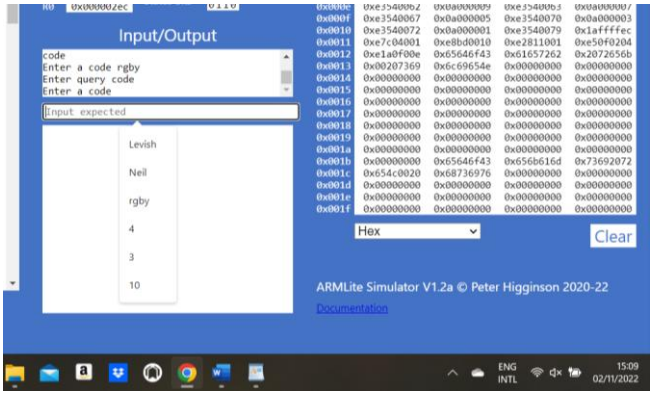
It loops back to the function, "getcode" to allow input of secret code and then, processing the conditions applied to the secret code.

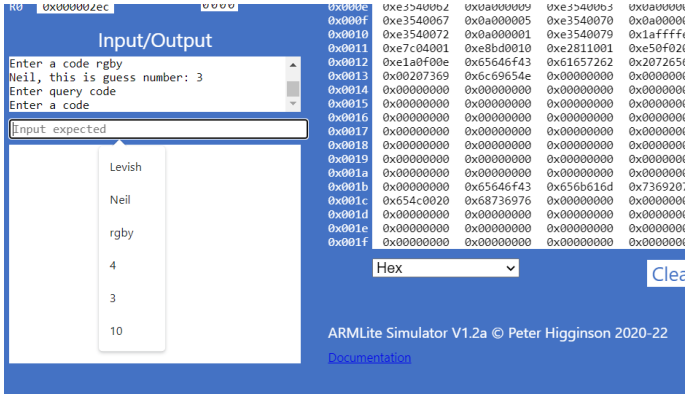


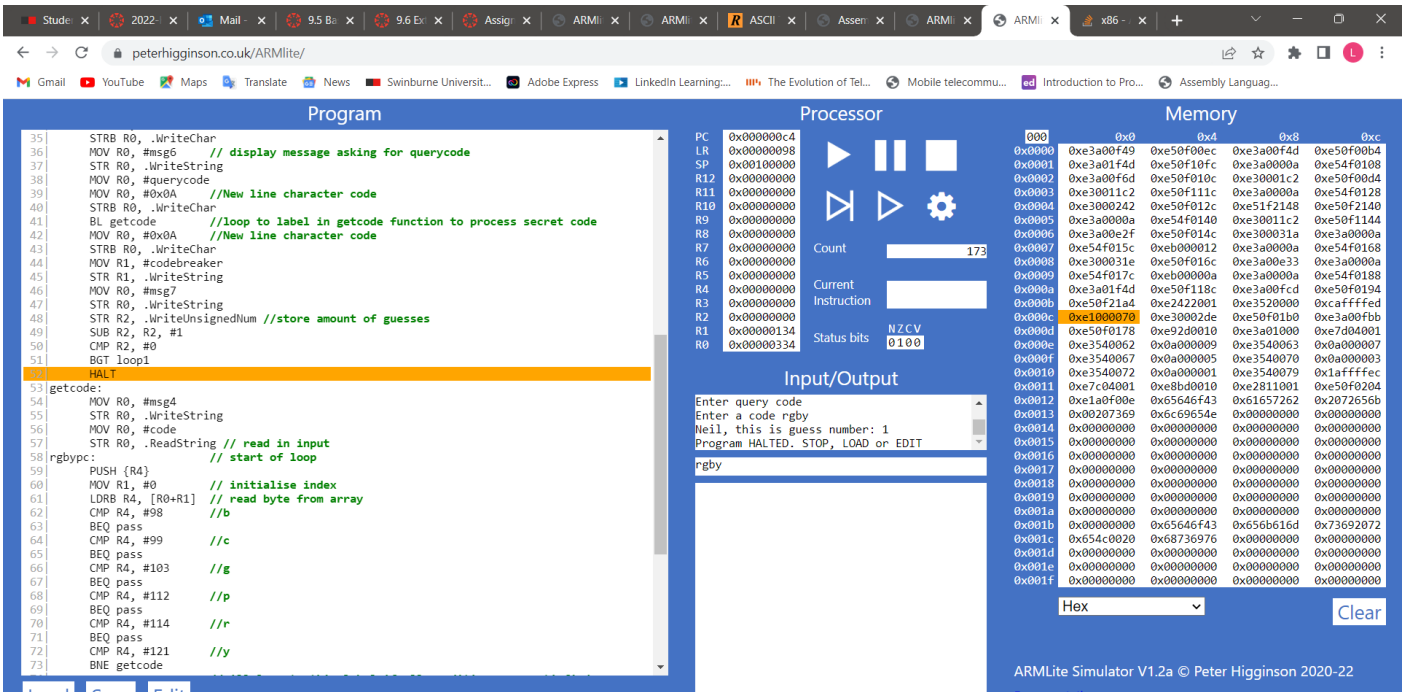
As the code is processed, after asking for the information to begin the name, it will ask for the codemaker to enter his/her secret code. Then, after displaying the command on the output box, the player will have to enter a secret code of the alphabet "rgbypc". After writing the code, the code will loop to the "getcode" function to process the structure of the code to see whether it satisfies the required and mentioned conditions. If it does not satisfy the conditions, it will keep on looping to the "getcode" function otherwise, it will keep on running till the code is halted.

Mastermind – Stage 4

Query Code Entry



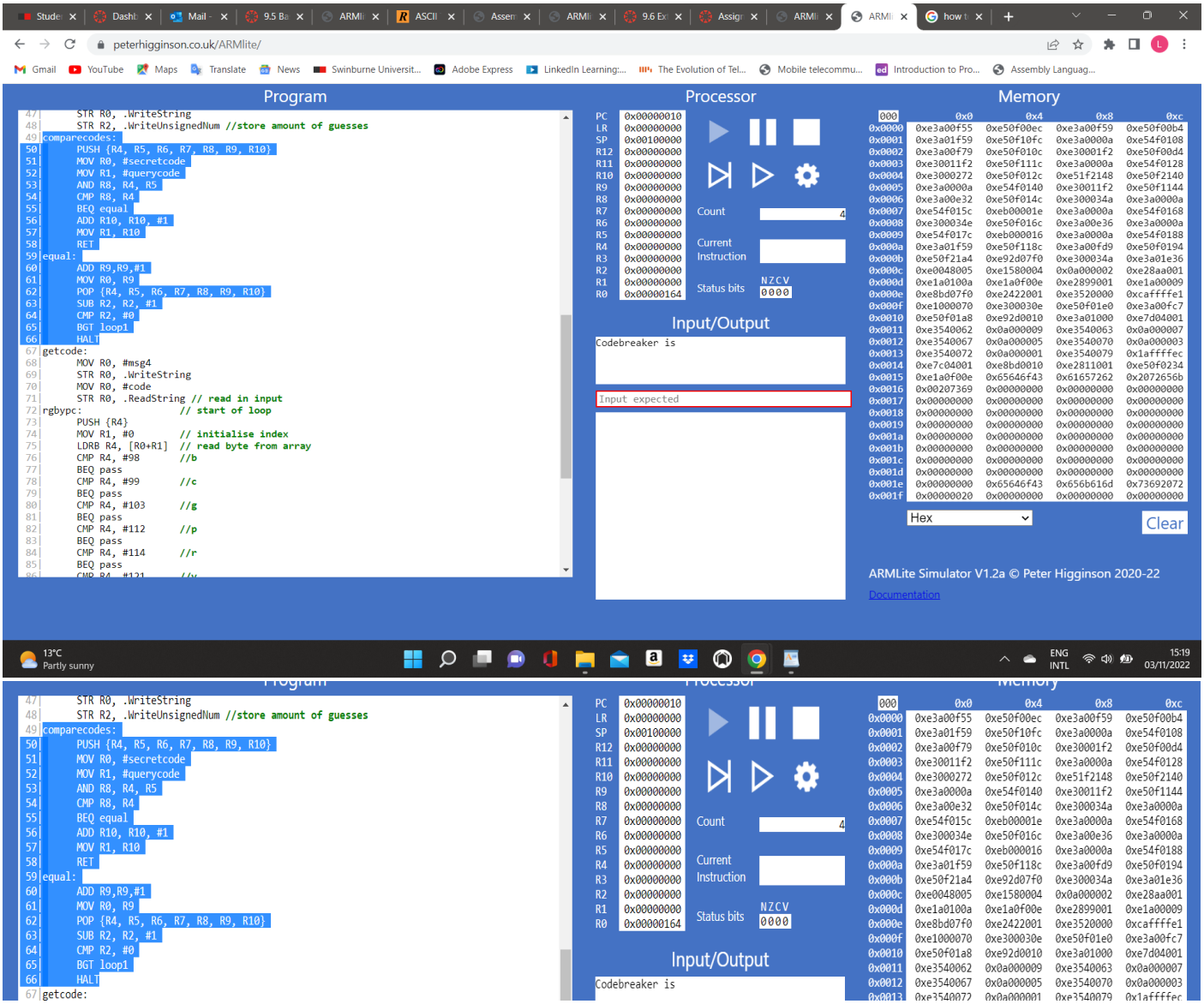




Just after the codemaker has entered his/ her secretcode, the game is on and the codebreaker has to figure out the same exact code that has been entered by his/her opponent. There the codebreaker get his/her chance to play next by getting to input a code followed after a command telling him/her to enter a query code. Now, to monitor the number of chances that the codebreaker has, we will use that dedicated register to keep account of the chances that are left. As the codebreaker enters a query code, he /she will get an alert, telling him/her how many chances are left and once the number of queries has reached its threshold, the program will halt.

Mastermind – Stage 5a

Query Code evaluation

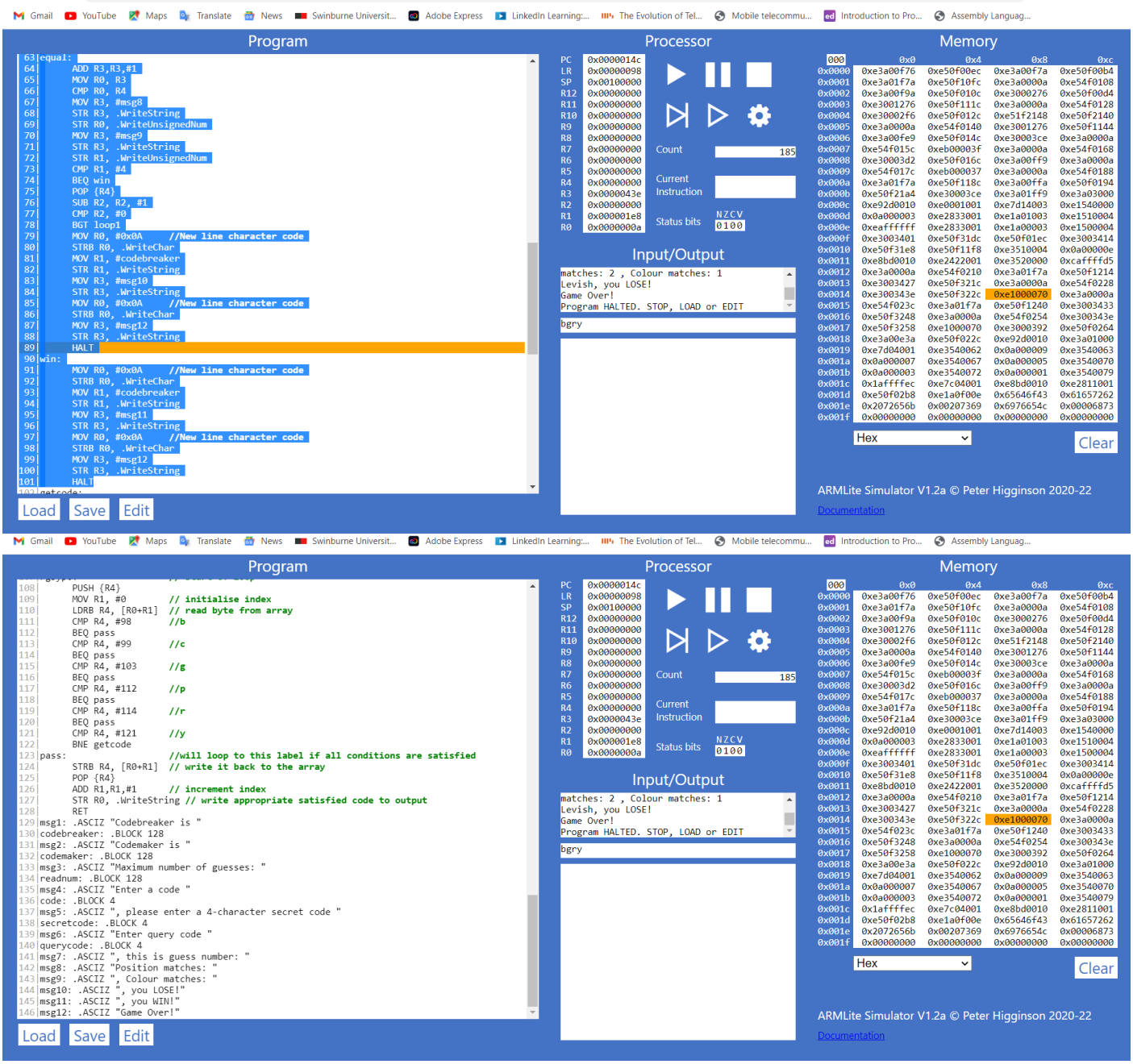



```
        STR R0, .WriteString
        STR R2, .WriteUnsignedNum //store amount of guesses
comparecodes:
        PUSH {R4, R5, R6, R7, R8, R9, R10}
        MOV R0, #secretcode
        MOV R1, #querycode
        AND R8, R4, R5
        CMP R8, R4
        BEQ equal
        ADD R10, R10, #1
        MOV R1, R10
        RET
equal:
        ADD R9,R9,#1
        MOV R0, R9
        POP {R4, R5, R6, R7, R8, R9, R10}
        SUB R2, R2, #1
        CMP R2, #0
        BGT loop1
        HALT
getcode:
        MOV R0, #msg4
        STR R0, .WriteString
        MOV R0, #code
        STR R0, .ReadString // read in input
rgbypc:                 // start of loop
        PUSH {R4}
        MOV R1, #0      // initialise index
        LDRB R4, [R0+R1]  // read byte from array
        CMP R4, #98     //b
        BEQ pass
        CMP R4, #99     //c
        BEQ pass
        CMP R4, #103    //g
        BEQ pass
        CMP R4, #112    //p
        BEQ pass
        CMP R4, #114    //r
        BEQ pass
```

decode that secret colored code until he/she wins or loses and this decision totally depends on the number of queries the codebreaker has been given. So, to determine the success or failure of the codebreaker, the secretcode and the querycode need to be compared and verified whether the color count and the position of the colored pegs are the same as the secret code. However, to do so, a function, (comaprecodes) has been added to compare all the mentioned attributes of the secretcode. There are various scenarios and this has been subdivided into case 1 and case 2 in the assignment itself. Therefore, case 1 will verify the number of colors which match, and it will be stored in R0. Case 2 determines and counts the number of pegs that are in the correct position and the count is stored in R1.

Mastermind – Stage 5b

Query Code evaluation



After implementing that function where it compares the query code and the secret code to count the number of pegs of the same color and that is positioned in the right order according to the attributes of the secret code, other lines are required to complete the code and the game. That is, the numbers have been stored in R0 and R1. Therefore, for the codebreaker to have an overview of his/her move about the order and color of pegs, the information stored in R0 and R1 need to be displayed and this must loop back to the querycode input every time the querycode does not match the secret code and thus determine whether the codebreaker win the game or lose. To do this, several displayed lines have been implemented to ensure a fair end of the game. This is Mastermind Game.