CT861 Assignment 1

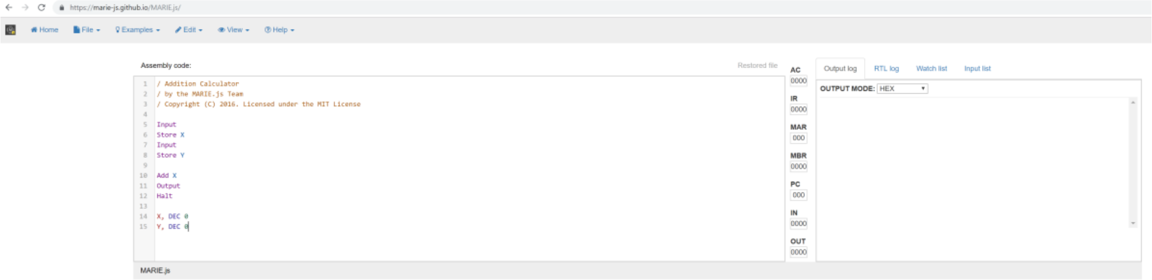
Assembler Programming using MARIE

Overview

The objectives of this assignment are as follows:

1. To reinforce your understanding of computer architectures and machine (assembler) languages.
2. to be able to write simple programs in Assembler code.

MARIE ('Machine Architecture that is Really Intuitive and Easy'), see Figure 1, is a simple (von Neuman) machine architecture and assembly language designed by Linda Null and Julia Lobur MARIE is quite similar to the example architectures demonstrated in previous lectures.



```
/ Addition Calculator
/ by the MARIE.js Team
/ Copyright (C) 2016. Licensed under the MIT License

Input
Store X
Input
Store Y

Add X
Output
Halt

X, DEC 0
Y, DEC 0
```

AC 0000
IR 0000
MAR 000
MBR 0000
PC 000
IN 0000
OUT 0000

OUTPUT MODE: HEX

Figure 1: The MARIE simulator (screenshot)

In this assignment you will study the MARIE architecture and write three programs using the MARIE Assembler language. It is an individual assignment with no formal labs. The MARIE simulator can be launched on your browser, and your assembler files can be locally stored / retrieved on your computer via the ‘Upload’ / ‘Download’ dialogues.
This assignment requires you to record a video explaining your solutions, therfore you need a screen recorder (e.g., Zoom) on your computer.
Please note that there will be a tutorial covering the MARIE assembler.

Problem 1: [2 marks]
Study the simulator software and the online documentation (https://marie-js.github.io/MARIE.js/book.pdf).
Explain in your own words the MARIE system architecture as shown on page 2, outlining the purpose of and interaction between all subsystems shown.

Problem 2: [2 marks]

Using the example code shown at the launch of MARIE as a starting point, write a simple program that:

- accepts user input for variables A, B, C, D,
- calculates the expression (C + 3 * A – (2 * B + D)), using the correct order of operations (i.e., BODMAS),
- stores the result in variable E, and
- prints the result on screen (via ‘Output’).

Fully explain / comment your code.

Hints:

1. Problem 2 is a simple expansion of the “Addition Calculator” example in MARIE with more variables.
   You can either implement the multiplication using a loop as shown in the “Multiplication Calculator” example, or simplify the expression, i.e.:

$$A + 4 * B = A + B + B + B + B$$

Problem 3: [6 marks]

Write a program that allows a user to iteratively (i.e. in a loop) enter positive numbers, until (s)he puts in a zero. The program will then show the biggest number that has been entered and terminate.

Fully explain / comment your code.

Hint:

1. Problem 3 is more complicated, and you should follow a structured approach, for example by executing the following steps:
   a. Pseudo-code description of the algorithm:

```
max = 0
value = 0

LOOP
    INPUT value

    IF (value == 0)
        EXIT LOOP
    ENDIF

    IF (value > max)
        max = value
    ENDIF

ENDLOOP
PRINT max
STOP
```

b. If needed draw a flowchart diagram to visualize the algorithm.
c. Examine the MARIE instruction set to see how the above code snippet can be mapped to a sequence of instructions.
   The tricky bits are the LOOP and the IF statements!
d. Implement the code incrementally, i.e. start with a single aspect, and make sure that it all works, before enhancing the code with the next feature. For example:
   i. Store user input in variable *value*, print it and halt.
   ii. Do i. in an endless loop, i.e. the program never terminates.
   iii. Terminate the loop if you enter a “0”, i.e. if *value* is zero (see also the *Skipcond (C)* instruction in the documentation)
   iv. Introduce the new variable *max* and assign *value* to *max*.
   v. Determine how you can compare *value* with *max*, and only update *max* if it is smaller than *value*.
      Hint: Calculate the difference between both and use *Skipcond (C)*.

Problem 4: [10 marks]
Write a program that calculates the binomial-coefficient “n over k” of the two entered integer values n and k (see https://en.wikipedia.org/wiki/Binomial_coefficient). Your program must verify that $n \geq k \geq 0$ before doing the calculation.
In your answer use (a hierarchy of) subroutines to implement the multiplication and factorial function needed.
Fully explain / comment your code.

Assignment Submission

Please submit a zipped folder to Canvas containing:

- 4 videos in which you explain your answers for problems 1, 2, 3 and 4. Identify yourself at the beginning of the video by name and student id.
- Your (well-commented!) source code for problem 2, 3 and 4 submitted as .mas files, which is MARIE’s file format for assembler listings. The headers of your solution must contain your name and student id.
  Please make sure that your programs terminate after completion!

Marking Scheme Outline

- Problem 1 [2 Marks]
  Full marks for comprehensive video explaining the MARIE architecture

- Problem 2 [2 Marks]
  1 mark for working code (up to 0.5 marks for attempt), 1 mark for video explanation.

- Problem 3 [6 Marks]
  3 marks for working and fully documented code (up to 1.5 marks for attempt), 3 marks for video walkthrough.

- Problem 4 [10 Marks]
  5 marks for working and fully documented code (up to 4 marks for attempt), 2 marks for video walkthrough.