CSE241 – OOP (Fall 2025)

Homework #2

Hand-in Policy: Source code and any documentation should be submitted online as a single .zip or .rar file with naming convention STUDENTID_LASTNAME_FIRSTNAME_H2.ZIP via Teams by the submission deadline. No late submissions will be accepted.

Collaboration Policy: No collaboration is permitted. Any cheating (copying someone else's work in any form) will result in a grade of -100 for the first offense and -200 for the subsequent attempts.
Grading: Each homework will be graded on the scale of 100. Unless otherwise noted, the questions/parts will be weighed equal.

You will continue implementing your CPU from the previous homework. This time, you will add some memory instructions. The memory addresses will be represented as number with a '#' prefix. For example, while "230" is a constant, "#230" is a memory address.

The set of instructions in HW1 are part of the CPU instructions. We add the following ones.

- Additional move instructions

  ```
  MOV reg, address
  MOV address, reg
  MOV address, constant
  ```

  For example `MOV R2, #16` copies the value of register 2 to memory location 16.

- Additional arithmetic instructions

  ```
  ADD reg, address
  SUB reg, address
  ```

  For example, `ADD R2, #16` adds the value of memory location 16 to register 1.

  Clarification: `SUB R1, R2` subtracts the value of register 2 from register 1 and puts the result into register 1.

- Additional jump instructions

  ```
  JPN reg, lineAdress
  ```

  For example, `JPN R2, #16` jumps to line 16 of the program if the value of R2 is zero or smaller.

- Additional input/output instructions

  ```
  PRN address
  ```

For example, `PRN #16` will print the value of the memory location 16 to the screen/console, after each print a new line should be inserted.

- Modified instructions

  `HLT`

  Halts the program and prints on the screen the contents of all registers as well as the memory.

Your program will run using command line parameters. The format for the command line parameters is as follows:

```
yourProg filename option
```

yourProg is the name of your executable file (as compiled by your C++ compiler), file name is the text file that contains your simple CPU instructions, option a number and the defines the how your program runs as follows

- option = 0: program will run and finish by executing each instruction.
- option = 1: program will execute each instruction after displaying the instruction first. It also will print the contents of all the registers.
- option = 2: your program will execute each instruction just like option = 1. This option will also print the contents of the memory after each instruction execution.

Important Considerations:

- Your memory can hold only 100 integers. Each integer is a single unsigned byte. Addresses start from 0.
- Your program can have at most 500 instructions or lines.
- Your program should handle error cases such as syntax errors in the input files. You should print an error message on the screen and halt the program if you detect an error in the input.
- With your submission, include the results of a few runs of your program with different programs and run options.
- Do not use any functions from the standard C library (like **printf**), you will use << and >> operators to print and write strings.
- You will use C++ string class to manipulate your strings. Do not use any string functions like **atoi** or **strtok** or similar. If you need these functions, then write your own!
- Your program should use object-oriented programming paradigms (i.e., user classes, encapsulate your data and algorithms as needed).
- Do not forget to “indent” your code and provide comments.