**REPORT**

# Implement CPU class

CPU class simulate a CPU and its registers, instructions, memory. CPU class has:

* 6 registers: private int PC, SP, IR, AC, X, Y;
* An array of int for memory: private int memory[];
* 2 stack pointers: USP (for user) and SSP (for system)
* mode to indicate the running mode (user or system)
* timer: to check with ticktock to start timer interrupt
* a list of constants for instructions

CPU class has:

* public void initializeMemory(int mem[]): initialize memory (this is loading a program to run)
* public int readMemory(int address): read from memory, interrupt if invalid range or no permission
* public void writeMemory(int address, int data): write to memory interrupt if invalid range or no permission
* public void fetchInstruction():Fetch instruction from memory to IR instruction is pointed by PC
* public void executeInstruction(): with type of instruction stored in IR, do the corresponding operations.
* public void run(): run the program, fetch each instruction then run it
* public void timerInterrupt(): to perform timer interrupt
* public void loadInstructions(int start, int instructions[]):to load a range of instructions
* public void push(int val): put value to stack (choose between system stack or user stack)
* public void pop(): pop from stack (system stack or user stack)
* public void pushRegisters(): save all registers (used before interrupt)
* public void restoreRegisters(): restore register (used after interrupt)
* public void indexOutOfRangeInterrupt(): this is to perform index out of range interrupt
* public void invalidAccessInterrupt():this is to perform invalid access interrupt

# Implement OSSimulator class

This class is to run simulator. Its arguments are the name of text file (where instructions are stored), the number for timer interrupt.

Parsing program file:

* Advoid empty line
* Check if a character at 0 is
  + “.”: Jump to address (right after the dot)
  + “-“: A negative number
  + Any other non-digit charater will be missed

After parsing the program file to load an array of instructions, it will set timer interrupt, initiate memory for CPU and then let CPU run.