OOD Fall 2020 Assignment 2: SIM part 1
Due: November 11th, 2020

You are asked to develop a C++ program to implement a simulator for a simple microprocessor. The microprocessor, called SIM (short for Simple Integer Machine), has two memory banks. The first, called instruction memory, contains 1024 memory locations, each capable of holding one instruction, and the second, called data memory, also contains 1024 memory locations, each capable of holding one integer value.

The SIM supports the following instructions:

Instruction	Format	Effect	remarks and addressing modes
ADD	ADD in1,in2,out	out := in1 + in2	in1 and in2 can either be addresses or constant. Out must be an address.
NEG	NEG in, out	out := - in	in can either be addresses or constant. Out must be an address.
MUL	MUL in1,in2,out	out := in1 * in2	in1 and in2 can either be addresses or constant. Out must be an address.
JMP	JMP a1	goto address a1	a1 must be a valid address within the instruction memory
JMP0	JMP0 in1, a1	if (in1 == 0) goto a1	in1 can either be addresses or constant. a1 must be a valid address within the instruction memory
ASS	ASS in1,out1	out1 := in1	in1 can either be addresses or constant. Out1 must be an address.
LE	LE in1,in2,out	if (in1 <= in2) then out :=1 else out := 0;	in1 and in2 can either be addresses or constant. Out must be an address.
READ	READ x	read an integer form keyboard device and store value in x	x must be a data address.
WRITE	WRITE x	print x on the display device	x can either be a constant or an address.
HALT	HALT	stop the SIM	

Your simulator should be capable of loading a program form a disk file, and executing that program. The simulator's user interface should be kept as simple as possible (i.e., no need for a GUI).

You should submit the following items:
(a) Class diagram showing the structure of the classes in your simulator;
(b) The documented C++ source code;
(c) Sample test programs to be run on your simulator; and
(d) A short users' guide describing how to build (compile and link) and use your simulator.