Functional Simulator for subset of RISC-V instruction Set

Directory Structure:

```
 |
 |- SOURCE CODES
    |
    |- PHASE1.cpp
    |- PHASE2.cpp
    |- opcode.txt
    |- data.txt
    |- operation.txt
 |- TESTS
    |
    |- bubble_sort.asm
    |- factorial.asm
    |- fibonacci.asm
 |- OUTPUTS
    |- mcode(sample).mc
    |- data_mem(sample).mc
    |- data_r(sample).mc
    |- output_bubbleSort
    |- output_fibonacci(5)
    |- output_factorial(7!)
 |- DOCS
    |- Design docs
    |- GROUP_README
```

How to execute

```
$ g++ PHASE1.cpp
$ ./a.out
$ g++ PHASE2.cpp
$ ./a.out
```

Assumptions

- Labels may be used before declaration of the variable. Also, access of a variable in another variable is also supported. For eg-
  Var1: .word 10
  Var2: Var1
  is supported (as in Venus)
- Assembly instructions are accepted without any commas.
  For example: add x1,x2,x3 will not be accepted ,while add x1 x2 x3 will be accepted.
- For instructions such as lw,sw,lh,sh etc the format is : lw x1,10(x2); i.e. brackets will be accepted.
- For jalr, the format is jalr x2 x1 2(as in Venus).
- The memory size is of 6000, starting at 0x10000000 and stack pointer starts from 0x10001770
- The values for immediate fields in u type, UJ Type, SB type and I type (except for load and store instructions) can be entered in binary ,decimal and hexadecimal formats.
  For eg: for an immediate value of 10, 0b1010, 10 , 0xa, all are accepted. Negative values are also accepted in these formats.

Errors Handled

- If **ld** or **sd** instructions are entered, they are considered as invalid.
- For branch or jump instructions , if an immediate of 0 is given, a message: “infinite loop. Enter finite offset ” is printed and code exits.
- If "lw x10 x20" or “sw x10 x20 ” or “x10 (x20)” is entered, “ no offset/immediate field entered” is printed as the error and the code exits .
- "ori x10, x20, x30" if given, is detected as an error.“ Immediate should be entered, not a register ” is printed and the code exits .
- If the immediate/offset value given in the instruction is beyond the limit,(such as (-2047-2048) for immediate values) “immediate out of range" printed code exits.
- If the register number of any of the source or the destination register is not in the range (0-31) then,”Invalid register number given” is printed and code exits..
- If the assembler encounters any unrecognised instruction, ‘Invalid Instruction!’ error is thrown and code exits.

- If there is an attempt to access memory outside the range 0x10000000-0x10001770, ‘Invalid memory access’ message is thrown, code exits.
- If a label is used, not declared in the entire code, “Label used but not declared” error is thrown. Also, if a label is declared twice, “Label declared twice!” error is thrown.
- If in .asciiz, string is not given as input, “Could not parse string!Error” is shown.Also not more than 2 strings can be stored in a single .asciiz statement(as in venus)
- If in branch statements, the offset is 0, “Infinite Loop!” error is thrown.
- If the number of parameter exceed or are less than required eg: add x10 x20 or add x10 x5 x5 x2 , then an error is detected and printed. The code exits in such a case
  .

Limitations

- Pseudo instructions not supported.
- Data memory is limited to 6000 bytes.
- Floating point numbers not supported.