Assignment 3
Deadline: 11:59PM on Apr 20

RISC V Assembly Language Program
You can work in a group of up to 4 for this assignment. Write a RISC V assembly language program which declares two variables for input and prints their values, memory addresses and sum of both variables without using a third variable. Follow the requirements for implementation.

Requirements
- Define two variables input1 and input2 with initial values of 5 and 3, variables should be declared following the .data identifier.
- Print variables and their memory addresses.
- Increase the value of both variables by 1 and print the new values.
- Add both variables without using a third variable and print the sum and memory address where you stored the sum.
- You can print a new line to distinguish outputs (optional).
- Use system call to print integers and addresses etc.
- Do not forget exit system call.

System call
In RISC V assembly language “ecall” instruction is used to invoke a system call. The specific system call number is passed to a specific register to indicate which system call a program is requesting. System call can be used to print the values by using a specific call number.

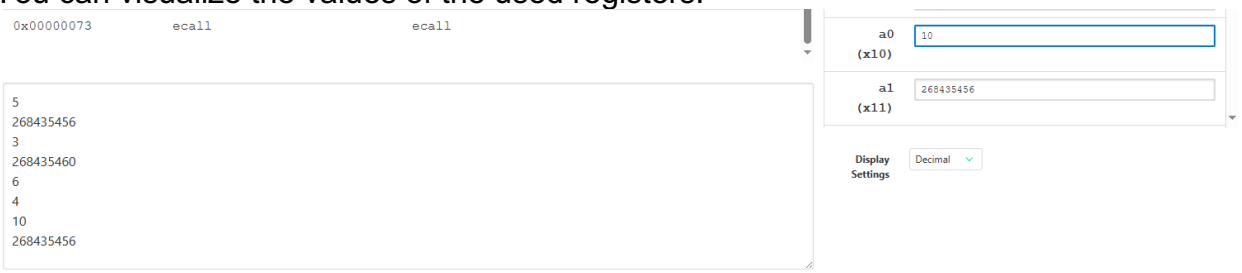
Example: Consider the code to understand the system call.

```
.data
        input: .word 5

.text
        .globl _start

_start:
        lw s2, input
        add a1, s2, x0
        li a0, 1 # System call number to print integer
        ecall
# print a new line
        li a0, 11
        li a1, 10
        ecall

li a0, 10 # System call number for exit
ecall
```

- The program uses system call number 1 (print integer) to print the integer stored in a0.
- The program exits using system call number 10 (exit).
- To print address in decimal, the system call number 1 can be used but you need to load address of the variable to a register using “la” instruction first.

Example

You can visualize the values of the used registers.



Instructions

- You can use venus risc v assembler simulator (https://venus.kvakil.me/).
- You can ignore any “Invalid ecall” in venus if you discover and use other system call numbers.

Grading

- (6 points) Correct solution to the problem.
- (4 points) Report with output screenshots and registers highlighted with value and address of result.

Submissions

riscv3.asm:

- Submit only one file riscv.asm which contains your assembly code. Include your name and student number as comments at the top of your code. You can copy the assembly code to a text editor and save as riscv.asm

a3_report.pdf:

- Submit a very short report which describes how to implement your solution and add a screenshot that shows your output memory address and address in the simulator highlighted.
- Include your name and student number in your report as well.

Put riscv3.asm and a3_report.pdf in a directory and compress the directory as a a3.zip file. Upload a single zip file.