Homework-1: CS250 & IN210 Assembly Language and Computer Organization, Rung-Bin Lin
March 16, 2022, Computer Science and Engineering; International Bachelor Program in Informatics, Yuan Ze University

Fibonacci Numbers

Due: March 16, 2022

Problem Description

Modify the RISC-V assembly program given in https://github.com/andrescv/Jupiter/blob/main/examples/fibonacci.s for computing Fibonacci numbers so that the modified program should only count the number of instructions **originally** being executed at each run. That is to say, any instructions added into the program should not be counted if they are executed. The number of instructions added should not exceed **60**. Note that you can only add instructions to the original code which should not be modified. Besides, you must run this program twice with different integers as input. Suppose the first given integer *j* results in a Fibonacci number F_j and the number of instructions executed is E_j. Similarly, the second given integer *k* results in a Fibonacci number F_k and the number of instructions executed is E_k. Calculate $(E_k - E_j)/(k-j)$ which is the average number of instructions executed per calculation of Fibonacci number. Also calculate $(F_k - F_j)/(k-j)$ which is the average growth rate (slope) of the Fibonacci numbers between the two given integers *j* and *k*.

You should use RISC-V instruction set simulator **Jupiter** (with I, F, M extensions) to develop and execute the assembly code. The Jupiter simulator can be downloaded from GitHub https://github.com/andrescv/Jupiter. To understand the code in Fibonacci.s, you need study how the assembler, directives, Ecalls, etc. are used to develop the assembly code. In particular, **Ecalls** implements a set of ABI's (Application Binary Interface) that can be used by the assembly code to interact with an operating system to read data from a keyboard and write data onto a monitor.

It is very easy to use Jupiter. Simply download the related file in its web site. Decompress the file and store the decompressed data in a new folder. Find a file named **Jupiter.bat** in the **bin** folder. Doubly click this file to invoke Jupiter. Then, you can write, assemble, execute, and debug the code directly on Jupiter.

Input Format

When the code is executed on Jupiter, Just follows the instruction presented on the monitor to provide input data.

Output Format

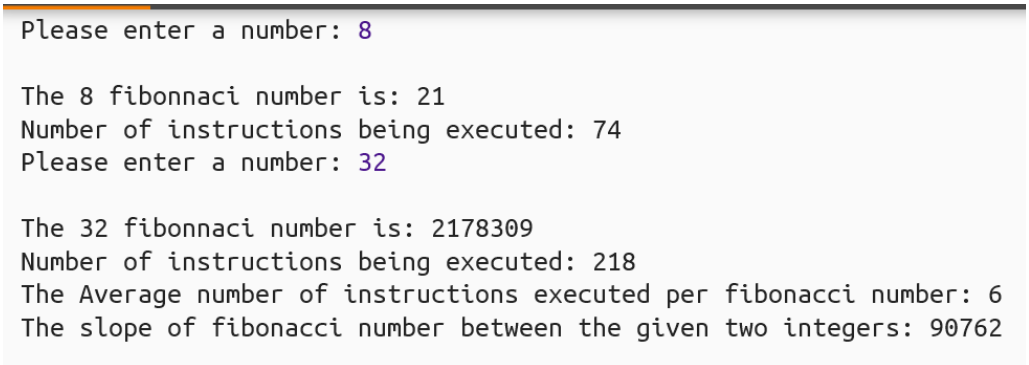
The output format should be exactly the same as the example given below. Certainly, different input will result in different numbers presented on the monitor.

What Should Be Handed In:

- Assembly code for modified Fibonacci.s. **The first line of assembly code should consist of your student ID number and your name**. Every **added** instruction should have a comment to explain what the instruction does. a comment should start with ### at the beginning of the comment. The file name of the assembly code should be **sID.s** where ID is your student ID number. A valid file name will look like s1091111.s .
- A clip like the one shown in the example of input and output below. Save the clip as a file called **sID.png**, where ID is your student ID number. A valid file name for an output clip will look like s1091111.png .
- The homework will not be graded if you do not follow the above rules.

Other information:
https://github.com/riscv-non-isa/riscv-asm-manual/blob/master/riscv-asm.md
https://shakti.org.in/docs/risc-v-asm-manual.pdf
https://github.com/andrescv/Jupiter
https://jupitersim.gitbook.io/jupiter/assembler/directives

Example 1 of Input and Output

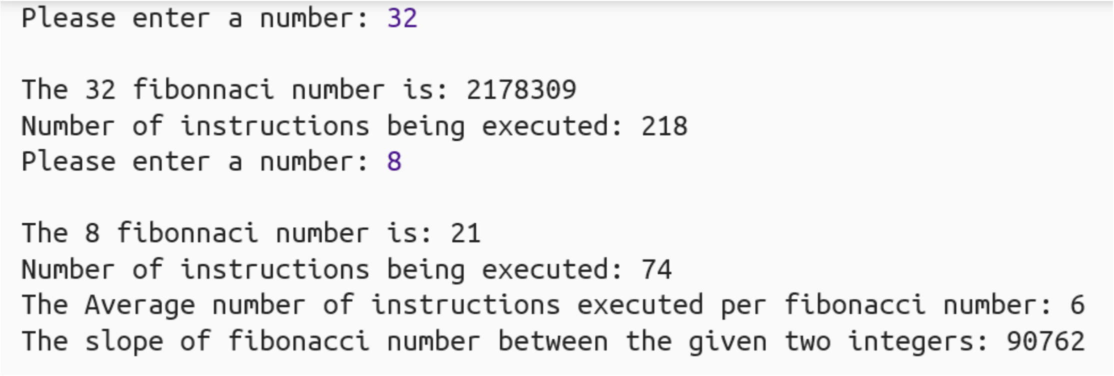


```
Please enter a number: 8

The 8 fibonnaci number is: 21
Number of instructions being executed: 74
Please enter a number: 32

The 32 fibonnaci number is: 2178309
Number of instructions being executed: 218
The Average number of instructions executed per fibonacci number: 6
The slope of fibonacci number between the given two integers: 90762
```

Example 2 of Input and Output

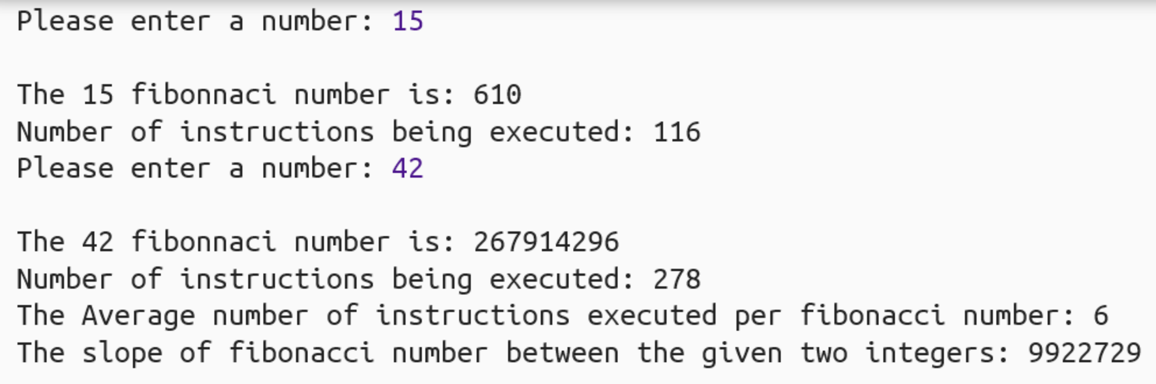


```
Please enter a number: 32

The 32 fibonnaci number is: 2178309
Number of instructions being executed: 218
Please enter a number: 8

The 8 fibonnaci number is: 21
Number of instructions being executed: 74
The Average number of instructions executed per fibonacci number: 6
The slope of fibonacci number between the given two integers: 90762
```

Example 3 of Input and Output

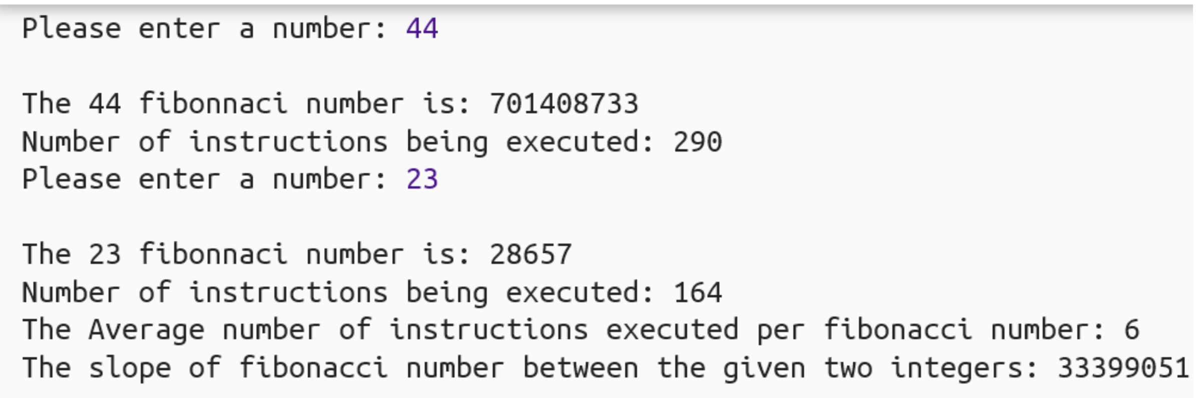


```
Please enter a number: 15

The 15 fibonnaci number is: 610
Number of instructions being executed: 116
Please enter a number: 42

The 42 fibonnaci number is: 267914296
Number of instructions being executed: 278
The Average number of instructions executed per fibonacci number: 6
The slope of fibonacci number between the given two integers: 9922729
```

Example 4 of Input and Output



```
Please enter a number: 44

The 44 fibonnaci number is: 701408733
Number of instructions being executed: 290
Please enter a number: 23

The 23 fibonnaci number is: 28657
Number of instructions being executed: 164
The Average number of instructions executed per fibonacci number: 6
The slope of fibonacci number between the given two integers: 33399051
```