Exercise 1

- Goal Getting familiar with the MIPS ISA
- Objective: Write an assembly program
- Platform: EduMIPS64 simulator
  - Download & documentation:

    http://www.edumips.org/
- Submission instructions: Bb

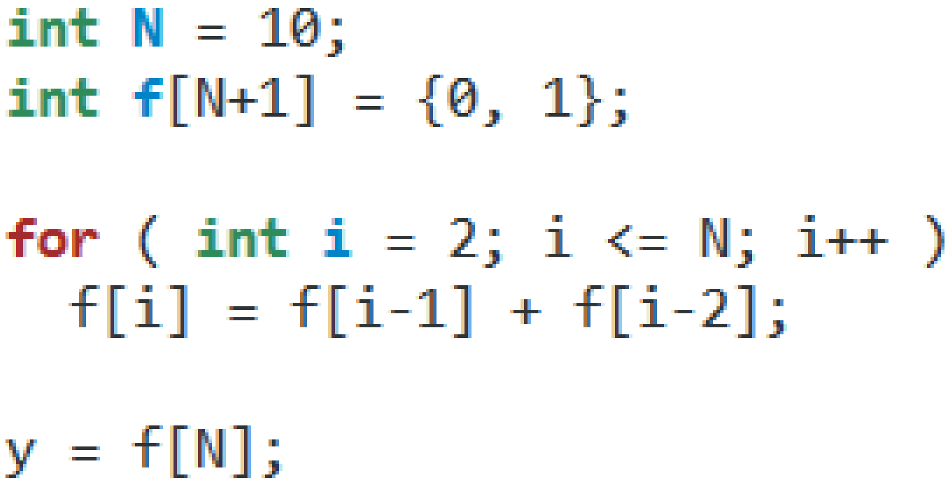
Fibonacci Numbers

- Fibonacci numbers, F_0, F_1, …, are defined as follows.
    - $F_0 = 0$
    - $F_1 = 1$
    - $F_n = F_{n-1} + F_{n-2}$
- For a given integer N (N > 1), print Fibonacci number F_N using iteration
    - – Input N is given at address 0
    - – Output F_N at address 8

Short list of instructions

- ADD rd, rs, rt ;; rd <- rs + rt (32-bit)
- DADD rd, rs, rt ;; rd <- rs + rt (64-bit)
- DADDI rd, rs, imm
- LD rt, offset(base) ;; rt <- M[offset + base], 64-bit
- LW rt, offset(base) ;; 32-bit, signed
- LWU rt, offset(base) ;; 32-bit unsigned
- SD rt, offset(base) ;; M[offset+base] <- rt, 64-bit
- SW rt, offset(base) ;; M[offset+base] <- rt, 32-bit
- SLT rd, rs, rt ;; rt = rs < rt
- SLL rd, rs, imm ;; rd = rs << imm

Algorithm?



```
int N = 10;
int f[N+1] = {0, 1};

for ( int i = 2; i <= N; i++ )
  f[i] = f[i-1] + f[i-2];

y = f[N];
```

Assembly Programming Language

- Details may vary depending on your tool chain (eg., asm.exe)
- Program contains more than just instructions
  - Instructions are called **text** or **code**
  - Directives (=commands)
    - .data
    - .text (=.code)
    - .org <n>
    - .word <n>, <n>, …
    - .word32 <n>, <n>, …
    - .byte <n>, <n>, …
    - .double <n>, <n>, …
    - .asciiz <s>
    - .align <n>