

CS-UY 2214 — Recitation 6

Introduction

Complete the following exercises. Unless otherwise specified, put your answers in a plain text file named `recitation6.txt`. Number your solution to each question. When you finish, submit your file on Gradescope. Then, in order to receive credit, you must ask your TA to check your work. Your work should be completed and checked during the recitation session.

You may consult the E20 manual, which is available on Brightspace.

Problems

1. Consider the following complete E20 program.

```
movi $1, foo           # ram[0]
lw $2, foo($0)         # ram[1]
halt                   # ram[2]
foo:
    .fill 42            # ram[3]
```

- (a) Assume that the initial value of all registers is 0. What will be the final value of the program counter, \$1, and \$2?
- (b) Convert this program into E20 machine code.

2. Consider the following complete E20 program.

```
movi $1, foo           # ram[0]
addi $1, $1, 1         # ram[1]
jr $1                  # ram[2]
halt                   # ram[3]
foo:
    movi $2, 2          # ram[4]
    movi $3, 3          # ram[5]
    halt                # ram[6]
```

- (a) Assume that the initial value of all registers is 0. What will be the final value of the program counter, \$1, \$2, and \$3?
- (b) Convert this program into E20 machine code.

3. Consider the following complete E20 program.

```
movi $1, 7
add $2, $1, $1
sw $1, foobar($0)
```

```

        sw $2, foobar($1)
        halt

foobar:
        .fill 600
        .fill 700

```

What memory addresses will be modified by this program, and what will their final value be?

4. Consider the following complete E20 program.

```

main:
    movi $1, 0          # ram[0]
    movi $7, 0          # ram[1]
repeat:
    lw $2, array($1)    # ram[2]
    jeq $2, $0, done    # ram[3]
    slt $5, $7, $2      # ram[4]
    jeq $5, $0, next    # ram[5]
    add $7, $0, $2      # ram[6]
next:
    addi $1, $1, 1      # ram[7]
    j repeat            # ram[8]
done:
    halt                # ram[9]
array:
    .fill 53            # ram[10]
    .fill 22            # ram[11]
    .fill 94            # ram[12]
    .fill 2             # ram[13]
    .fill 19            # ram[14]
    .fill 0             # ram[15]

```

Assume that the initial value of all registers is 0. What will be the final value of \$7? In English, what does this program do? Discuss how it works.

5. Consider the following complete E20 program.

```

foo:
    movi $1, foo        # ram[0]
    lw $2, foo($0)      # ram[1]
    halt                # ram[2]

```

Assume that the initial value of all registers is 0. What will be the final value of the program counter, \$1, and \$2?

Hint: compare this question to question 1.

6. Consider that we have two unknown values in labeled memory locations:

```

first: .fill somenumber
second: .fill anothernumber

```

Write a program that will swap the values of these cells. That is, when your program ends, the value *anothernumber* will be in the cell identified by the label **first**; and the value *somenumber* will be in the cell identified by the label **second**.

7. You are given a partial E20 program containing only the following:

```
beginning:
    .fill 3
    .fill 1
    .fill 0
    .fill 9
    .fill 5
    .fill 7
    .fill 1
    .fill 4
end:
```

That is, your program's data consists of an array of numbers between addresses **beginning** (inclusive) and **end** (exclusive).

Complete the program so that it reverses the contents of the array in place. In other words, when the program ends, the value at memory address **beginning** will be 4; the value at memory address **beginning+1** will be 1; the value at **beginning+2** will be 7; the value at memory address **end-1** will be 3; etc.

You must implement your program using a loop. Your program should work with an array of any length. You may not assume that the length or content of the array is the same as given above. The only thing you know about the array is that it is bound by labels **beginning** and **end**.

Hint: use paired **lw** and **sw** to copy a memory value from one address to another.