Homework 12

Homework 12 is due on Wednesday, May 6, at the start of the lecture. Remember to include your *Discussions section* (e.g. ED1) and follow the complete Homework submission guidelines.

Please ask all questions about this assignment during the office hours, or post them on piazza.

Programming in binary machine language

For your programs: write the instruction address, the binary LC-3 instruction, and the corresponding **RTL** or assembly in the following format:

```
0001 001 010 1 00100          ; R1  R2 + 4
0001 001 010 1 00100          ; ADD R1, R2, #4
```

Note how **bits are arranged in groups that correspond to instruction-specific operands**.

If you are not sure about the RTL format of a certain instruction, you can refer to Appendix A of your Patt & Patel text or read sections 5.5 and 6.1. Try to make your program as simple and intuitive as you can.

1. Register manipulation

1. Copy a value from register R2 into register R3 using
    a. Only one ADD instruction
    b. Only one AND instruction
2. Clear value of register R5 using only one instruction
3. Using only one instruction, increment value in register R6 by 1.
4. Using only one instruction, decrement value in register R5 by 2.

2. Memory addressing

1. Using only one instruction, read value from memory located 20 memory locations away from the address stored in PC.
2. Using only one instruction, generate and store in R5 address of a memory location 20 memory locations away from the current PC. Then demonstrate how the address stored in R5 can be used to access memory at that address.

3. Patt & Patel 6.4

Systematically decompose the problem 6.4 from Patt & Patel textbook to the level of LC-3 instructions, then write LC-3 instructions to implement your solution. You may change the contents of R0, R1, and R2. Turn in your flow chart and LC-3 instructions in binary. For credit, each instruction must be annotated with a comment in RTL or assembly.

4. 2^n

Assuming that R4 contains a positive value less than 15, put the value 2^{R4} into R3. You may change the contents of R3 and R4. Systematically decompose the problem to the level of LC-3 instructions, then write LC-3 instructions to implement your solution. Turn in your flow chart and LC-3 instructions in binary. For credit, each instruction must be annotated with a comment in RTL or assembly.

5. Sum of sequence

Assuming that R5 contains a strictly positive number, compute the sum of integers from 1 to R5 and store the result in R2. You may change the contents of R2 and R5. Systematically decompose the problem to the level of LC-3 instructions, then write LC-3 instructions to implement your solution. Turn in your flow chart and LC-3 instructions in binary. For credit, each instruction must be annotated with a comment in RTL or assembly.

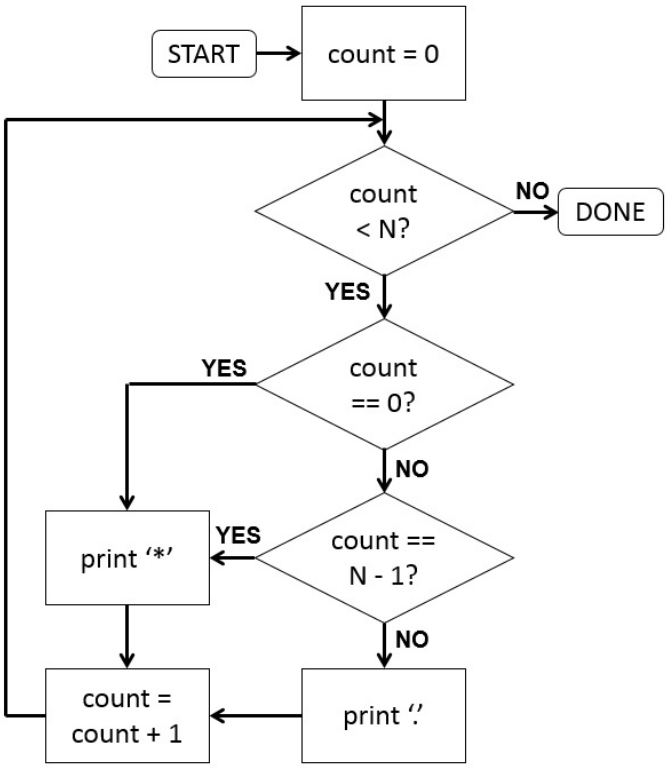
6. Patt & Patel 6.16

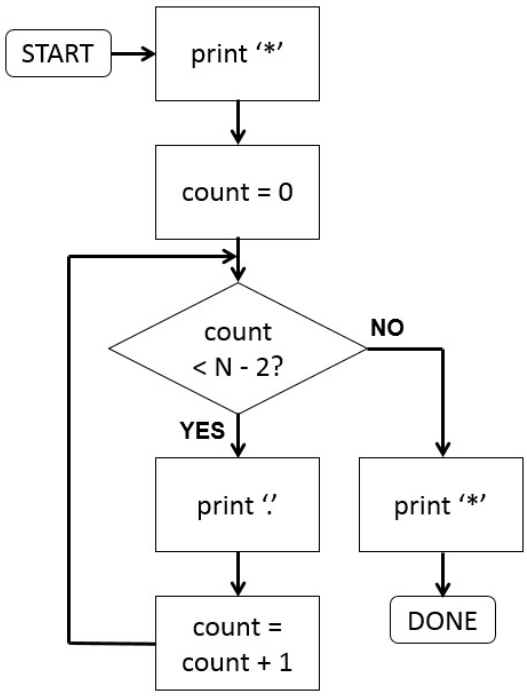
Solve problem 6.16 from Patt & Patel. Stop your execution trace when the PC reaches x3003, and do not fill in row x3003 of the table in your solution.

7. Printing a line

Two students are trying to print a line of periods bounded by asterisks to the console. The line is supposed to have N – 2 periods, where N 2. For example, when N = 8, they want to print, "***.....***" to the console (without quotes). Each student came up with a systematic decomposition that appears very different from the other student's solution. Look at the two approaches, as represented by the flow charts below.

1. Explain which approach is the better of the two and why you believe it to be better.
2. Imagine that one must replace each box labeled "print '*'" with a complex algorithm requiring almost 1,000 LC-3 instructions to implement. Repeat your comparison between the two approaches after the replacement described, again explaining why you believe your choice to be the better one.


START
count = 0
count
< N?
NO
DONE
YES
YES
count
== 0?
NO
print ‘*’
YES
count ==
N - 1?
NO
count =
count + 1
print ‘.’


START
print ‘*’
count = 0
count
< N - 2?
NO
YES
print ‘.’
print ‘*’
count =
count + 1
DONE