CS 429 Homework 5

Name: ______________________ Section #: ______________________

Note that this assignment assumes you have been doing the reading in Chapter 4 of the textbook.

Instructions: Type in the answers to the questions below and submit on Canvas. As usual, you may collaborate with your classmates and ask for assistance from the TA. But don't copy anyone else's answer. Each problem is worth the same number of points (more or less).

1. Show the machine code corresponding to the Y86 version of the `sumInts` program from Slideset 6. Assume that the code starts at location 0x050. For each statement, list `address:  code bytes`.

2. Annotate the following Y86 code (add comments to each line) explaining what the line accomplishes.

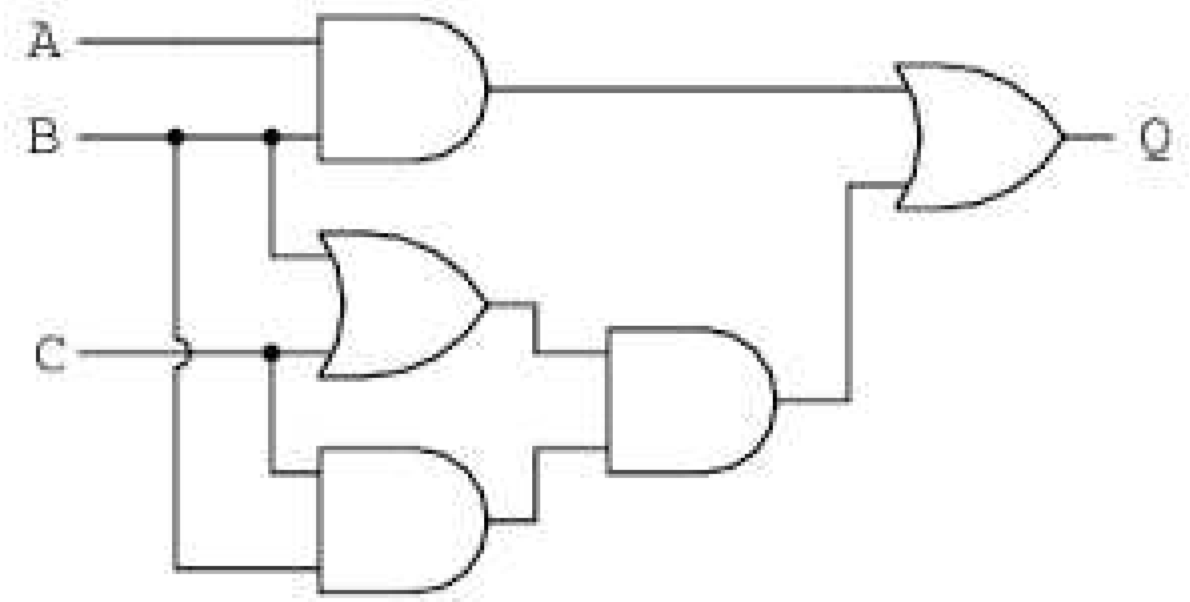
```
proc:
   irmovq   $0, %rax
   irmovq   $1, %r9
   rrmovq   %rdi, %r8
   andq     %r8, %r8
   je       done
loop:
   addq     %rsi, %rax
   subq     %r9, %r8
   andq     %r8, %r8
   jne      loop
done:
   ret
```

3. Explain what function is computed by the code above. Are there any constraints on the parameters? (Will it work for arbitary integer inputs?)

4. Write a C function that might compile into the above code.

5. Assuming that `proc` is at location 0x100, determine the Y86 byte encoding for the Y86 program. For each statement, list `address:  code bytes`.

Questions continue on the back.



6. Consider the circuit above. First, write a C expression E that describes the output Q in terms of the three inputs, A, B, and C. Then, write, compile, and execute a C function that prints the truth table for this circuit, including the outputs of each of the intermediate gates (call them W, X, Y, and Z from top to bottom, then left to right). *Use your expression E to generate the outputs; i.e., don't precompute them and just print the constants.* Loop through the input value; i.e., don't just call the expression 8 times. In fact, E should only appear once in your code, inside the loop body. Submit your code and the output, which should be similar to the following:

```
 A B C | W X Y Z | Q
--------------------
 0 0 0 | 0 0 0 0 | 0
 0 0 1 | 0 1 0 0 | 0
 0 1 0 | 0 1 0 0 | 0
 0 1 1 | 0 1 1 1 | 1
 1 0 0 | 0 0 0 0 | 0
 1 0 1 | 0 1 0 0 | 0
 1 1 0 | 1 1 0 0 | 1
 1 1 1 | 1 1 1 1 | 1
```