

# CS-UY 2214 — Recitation 1

## Introduction

Complete the following 10 problems. Put your answers in a plain text file named **recitation1.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.

Please note that your solutions must be in a plain text file. Other formats, such as PDF, RTF, and Microsoft Word, will not be accepted. Here are some recommended editors that produce plain text files:

- Notepad (comes with Windows)
- TextEdit (comes with Mac OS); note that if you are using TextEdit, you need to select “Make Plain Text” from the Format menu before saving the file
- gedit (available on most Linux distributions)
- nano (available on most Linux distributions)
- Sublime Text
- VSCode
- Atom
- Vim
- Emacs

For questions that require a solution expressed as an image, submit the image as a separate file. The image file should be named **recitationnqm**, where n is the recitation number and m is the question number; use an appropriate suffix (either jpg or png).

## Problems

1. Design a circuit that takes three inputs, A, B, and C, and outputs 1 if and only if exactly two of the inputs are 1. [Diagram needed here - **recitation1q1.jpg**] Express the circuit as a Boolean expression, using only AND, OR, and NOT. Provide a truth table for this circuit.
2. Using only AND, OR, and NOT gates, construct a circuit diagram that will calculate a NAND gate. Your circuit will have two inputs A and B, and one output Y. Submit your answer as an image, in accordance with the instructions at the beginning of this document. Your image may be a diagram created in a drawing program, or it may be a photograph of a hand-drawn diagram on paper. [**recitation1q2.jpg**]
3. Convert the following decimal numbers into binary. (a) 42 (b) 127 (c) 255
4. Convert the following binary numbers into decimal. (a) 1110 (b) 0101 (c) 1000000
5. Convert the following decimal numbers into hexadecimal. (a) 25 (b) 100 (c) 2047
6. Convert the following hexadecimal numbers into binary. (a) 1a2b (b) cafe (c) 0xdeadbeef
7. Convert the following decimal numbers into 8-bit binary 2's complement. Write all the digits. (a) 67 (b) 67 (c) 128
8. Convert the following 8-bit binary unsigned numbers into decimal. (a) 01101100 (b) 10000001 (c) 00000000
9. Convert the following 8-bit binary 2's complement numbers into decimal. (a) 01101100 (b) 10000001 (c) 11111111
10. Consider the following C++ program. Do not run it.

```
#include <iostream>
using namespace std;
int main () {
    int i = 1024;
    int count = 0;
    while (i > 0) {
        i /= 2;
        count++;
    }
    cout << " Completed " << count << " iterations " << endl;
    return 0;
}
```

Your friend Rufus argues that this program will correctly count the number of times it can divide 1024 by 2 before reaching zero. Is he right or wrong? Justify your opinion with a persuasive explanation, but do not type in or run the program. Predict the output of the program.