CS-UY 2214 — Recitation 1

Introduction

Complete the following exercises. 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. Consider the following circuit. (Image required: `recitation1q1.jpg` or `recitation1q1.png`)

   This circuit has inputs A, B, and C, and output Y. 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 NAND. 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. (Image required: `recitation1q2.jpg` or `recitation1q2.png`)

3. Convert the following decimal numbers into binary.

   (a) 42

   (b) 127

4. Convert the following binary numbers into decimal.

   (a) 1110

   (b) 0101

5. Convert the following decimal numbers into hexadecimal.

   (a) 255

   (b) 102

6. Convert the following hexadecimal numbers into binary.

   (a) 1a2b

   (b) cafe

7. Convert the following decimal numbers into 8-bit binary 2's complement. Write all the digits.

   (a) 67

   (b) 67

8. Convert the following 8-bit binary unsigned numbers into decimal.

   (a) 01101101

   (b) 10000000

   (c) 00000000

9. Convert the following 8-bit binary 2's complement numbers into decimal.

   (a) 01101101

   (b) 10000000

   (c) 11111111

10. Consider the following C++ program. Do not run it.

```
#include <iostream>
using namespace std;
int main () {
int i = 10;
int count = 0;
while (i > 0) {
i -= 2;
count++;
}
cout << "Completed " << count << " iterations " << endl;
return 0;
}
```

    Your friend Rufus argues that this program has an infinite loop, and will therefore never end. 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.

11. For this class, you are required to use a Linux programming environment. Take this opportunity to set up such an environment. Use one of the options enumerated in the syllabus: Anubis, Vital, VirtualBox, WSL, or a proper Linux installation. Log in to your Linux environment. Then, submit a screenshot of your working Linux programming environment. The screenshot should simply show the Linux desktop. You don't need to do anything after you log in. (Image required: `recitation1q11.jpg` or `recitation1q11.png`)

12. The course presumes a satisfactory knowledge of programming. In addition, we want to know that you are able to create, edit, and run programs in a Linux environment. Demonstrate that you can do programming under Linux by following these instructions. Your TA will help you if you get stuck. You will need to complete 7 problems.

    (a) Log in to your Linux environment: Anubis, Vital, VirtualBox, WSL, or a proper Linux installation.

    (b) Open a text editor on your Linux environment.

(c) Save your new file as either $\texttt{sum}_s eries.cpp(for C++) or$

(c) Write a program that calculates the sum of the series 1 + 2 + 3 + ... + n, where n is an integer input by the user.

(d) Open a Terminal.

(e) Run your program. Ensure the program handles potential errors such as non-numeric input gracefully.

(f) Take a screenshot of the Terminal showing the execution of your program with at least two test cases (one positive integer and one non-numeric input). Submit this screenshot. (Image required: recitation1q12.jpg or recitation1q12.png)