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 (specifically, the last question), submit the image as a separate file. The image file should be named `recitation`n`q`m, where n is the recitation number and m is the question number; use an appropriate suffix (either `jpg` or `png`).

Problems

For these exercises, do not use a calculator. You should perform the calculations yourself.

1. Convert the following decimal numbers into binary.
   (a) 35
   (b) 64
2. Convert the following binary numbers into decimal.
   (a) 1001
   (b) 1010101

3. Convert the following decimal numbers into hexadecimal.
   (a) 35
   (b) 64
   (c) 255

4. Convert the following hexadecimal numbers into binary.
   (a) f00f
   (b) abcd
   (c) 8888

5. 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.

6. We want to know that you are able to create, edit, and run programs in a Linux environment. For this class, we expect that all students are experienced with Python and/or C++. Demonstrate that you can work with these languages under Linux by following these instructions. Your TA will help you if you get stuck.
   (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.
       - If you're using Anubis, simply select New File from the File menu to open a new text editing window.
       - If you're using Vital or VirtualBox, you should have a program called Text Editor or `gedit` available.
       - In any case, ask your TA if you can't find the right program.

       Save your new file as either `hello.cpp` (for C++) or `hello.py` (for Python). Make sure that you save the file in your *home directory*, which will typically be `/home/anubis` on Anubis, or `/home/ubuntu` on Vital.
   (c) Use the text editor to write a program in either Python or C++. Your program should simply print out the message `Hello, CompArch!` Everyone should be able to write this program.
   (d) Open a Terminal in your Linux environment.
       - If you're using Anubis, select New Terminal from the Terminal menu to open a command prompt window.
       - If you're using Vital or VirtualBox, you should have a program called Terminal available.
       - In any case, ask your TA if you can't find the right program.
   (e) Run your program, after compiling it if necessary.
       - To run your Python program, type `python3 hello.py` and press enter. Your program should run. If you see an error message, make sure that your program is saved is the right place.
       - To compile your C++ program, type `g++ hello.cpp -o hello` and press enter. If compilation is successful, you will see no error message, and a new executable file named `hello` will be created by the compiler.
         To run the program, now type `./hello` and press enter. Your program should run. If you see an error message, make sure that your program is saved is the right place.
       - In any case, ask your TA if you're having trouble.
   (f) Take a screenshot of the Terminal showing the execution of your program. Submit this screenshot.