

Quiz 1 - Wednesday

July 3, 2024

Instructions

- You have time to finish the quiz before the lab session is over. **No need to submit anything for these quizzes**, but you **must** show your codes and results to your TAs before you leave the class. TAs may ask you questions and request you to work with your codes, e.g. asking you to run and compile, add/remove, and debug your program. TAs will provide you with feedback and your grade right away, but the grades will be released on Avenue a few days later.
- During the quizzes using AI generative models, like ChatGPT, is **NOT** allowed, as I believe quizzes are easy enough to be handled. But you can search on the internet, take a look at the lecture notes, talk to your friends, or even having your TA's help.
- The quiz starts at the beginning of the lab sessions, and only those who are present in the class in person can take the quiz. Please don't be late and make sure you are in class 15 min before the quiz starts.

1. Basic Commands (1 points)

- (a) Begin by downloading the provided 1XC3.zip and unzip it to any directory.
- (b) After unzipping, enter the 1XC3 folder, and run the `tree` command. If you do not have it on Linux you will see:

```
Command 'tree' not found, but can be installed with:  
sudo snap install tree # version 1.8.0+pkg-3fd6, or  
sudo apt install tree # version 2.0.2-1  
See 'snap info tree' for additional versions.
```

Install it first using: `sudo apt-get install tree` (Linux) or `brew install tree` (MacOS). The overall file structure looks like this:

```

pedram@pedram-GL553VE:~/COMPSCI1XC3/homework_quiz/1XC3$ tree
.
├── labs
│   ├── lab1
│   ├── lab2
│   │   └── lab2.txt
│   ├── lab3
│   ├── lab4
│   │   └── HelloWorld.jpg
│   ├── lab5
│   ├── lab7
│   └── extra_file.c
7 directories, 3 files
pedram@pedram-GL553VE:~/COMPSCI1XC3/homework_quiz/1XC3$

```

Figure 1: Terminal output after `tree`.

(c) Use the commands in the following table to finish the rest of the tasks.

Command	Description
<code>ls <file path></code>	Display the contents of the directory
<code>cd <file path></code>	Enter the directory
<code>cd ..</code>	Ascend one directory in the file system hierarchy
<code>pwd</code>	Display the current directory
<code>mkdir <file path></code>	Create a new directory
<code>touch <file path></code>	Create a file
<code>cp <file path> <file path></code>	Copy a file from into
<code>mv <file path> <file path></code>	Move a file from into
<code>rm -rf <file path></code>	Delete a file from
<code>nano <file path></code>	Edit a file from

(d) Create a file named `main.c` in `lab1` and write the following code into the file. Compile the code and share your results.

```

#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}

```

(e) Copy and paste `lab2.txt` from `lab2` to `lab3` and rename the file to `lab3.txt`.

(f) Move `HelloWorld.jpg` from `lab4` to `lab5`.

(g) Create a new folder named `lab6` inside `labs`.

(h) Delete `lab7`. Example output:

```
pedram@pedram-GL553VE:~/COMPSCI1XC3/homework_quiz/1XC3$ tree
.
├── labs
│   ├── lab1
│   │   ├── main
│   │   └── main.c
│   ├── lab2
│   │   └── lab2.txt
│   ├── lab3
│   │   └── lab3.txt
│   ├── lab4
│   │   └── lab5
│   │       └── HelloWorld.jpg
│   └── lab6
└──
7 directories, 5 files
pedram@pedram-GL553VE:~/COMPSCI1XC3/homework_quiz/1XC3$
```

Figure 2: Terminal output after making changes.

2. Limit and Overflow (0.5 points)

In lectures, we discussed the limits for various data types and their boundaries. Write a program which adds two variables `uint8_t a` and `uint8_t b`. Then print the addition of `a` and `b` which is equal to `uint8_t c`.

- What is the output when $a = 10$ and $b = 100$?
- Run the program when $a = 100$ and $b = 250$.
 - Why is the addition of a and b incorrect?
 - How can you change the code or conditions to get the right answer?

Keep your C codes for this question in `q2_hw1.c` file.

3. Characters and Strings (0.5 points)

Write a C program (name it `q3_hw1.c`) and given the string “FOOBAR” perform the following tasks using functions found in the `string.h` library:

- Find and print the length of the string.
- Copy the first three characters from the string into a new string and print the new string.

4. More Shell Commands (1 points)

This is the content of the file `lab3.txt` from the first question:

```
Zylophont reaxar galactic zarnith
Translate the xylophont for a universal code
Mysterious codes encrypted in the cosmic dust
Xylophont lexicon: crion, zeltron, vornax
```

```
The ancient prophecy speaks of the xylophont awakening
Decrypt the cosmic signals to unveil the interstellar truth
Translate xylophont messages for peaceful communication
In the cosmic symphony, the xylophont echoes resonate
Xylophont harmonies connect galaxies in a celestial dance
Interpret the xylophont glyphs to understand their cosmic language
Unravel the mysteries encoded in the alien scripts
Discover the intergalactic wisdom hidden in the xylophont verses
The xylophont whispers carry knowledge across the vastness of space
Decipher the celestial symbols for a glimpse into extraterrestrial intelligence
```

We want to perform a series of bash commands. Search on the internet or ask your TAs to find the shell commands to:

- (a) Display the first 10 lines of the file to get a glimpse of the encoded message.
- (b) Count the number of words in the file.
- (c) Search for occurrences of the word "translate" in the file.
- (d) Extract all lines containing the word "translate" and save them to a new file called `translation_clues.txt`.
- (e) Use a command to replace all occurrences of the word "Xylophont" with its English translation "Greetings" in the original file.

Tips You may need depending on you OS the shell commands like `head`, `wc`, `grep` and `sed`.

You can keep the results in Terminal opened to show the outputs to you TA.

5. Prints and Placeholders Tasks (1 points)

Consider the following C program:

```
#include <stdio.h>

int main() {
    int num1 = 10;
    int num2 = 20;
    char letter = 'A';
    char name[] = "John";

    // Task 1
    printf("Task 1: \n");
```

```
// Your Task 1 code goes here

// Task 2
printf("\nTask 2: \n");
// Your Task 2 code goes here

// Task 3
printf("\nTask 3: \n");
// Your Task 3 code goes here

// Task 4
printf("\nTask 4: \n");
// Your Task 4 code goes here
}
```

Complete the following tasks by adding appropriate placeholders in the `printf` statements:

- (a) Print the values of `num1`, `num2`, `letter`, and `name` using placeholders.
- (b) Update `num1` to `15` and print the updated value.
- (c) Concatenate the `letter` and `num2` and print the result.
- (d) Print the ASCII value of the `letter`.

Keep your C codes for this question in `q5_hw1.c` file.

6. Random Integer (1 points)

Search on the internet and write a C code to create and print a random integer number between 20 to 50. You **must**

1. compile and
2. run

the program **only** on **VScode** while showing the procedure to your TA.

Keep your C codes for this question in `q6_hw1.c` file.