

Quiz/Homework 1

The questions are designed by the TAs. For information on how to contact the TAs, please refer to the course outline.

Due Date: **From** Monday Jan 29th to Friday Feb 2nd during your labs sessions.

Instructions

- Please ensure that you complete all questions and present your answers to your TAs during **one** of your **assigned** weekly lab sessions for this week. TAs may ask you questions and request you to work with your codes, following the guidelines outlined in the Grading Policy section of the course syllabus regarding the presentation of your coursework. This session is referred to as a **Quiz**, during which TAs will provide feedback and your grade immediately. Grades will also be released on Avenue a few days later.
- The purpose of these Quiz/Home work assignments is not solely to test your programming skills. Instead, they are designed to provide you with simple and basic examples to better prepare you for your assignments. Consequently, you can work on the questions outside of the lab sessions. You may ask questions from you TAs at any point over Teams or even during the lab sessions. This is why we call it **Homework**.
- Please attend only the labs for which you are enrolled to prevent classes from becoming overcrowded.

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 Table to finish the following tasks:
- (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`.
- (i) 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, you have learned about limits for various data types and their associated boundaries. Write a program which adds two variables `uint8_t a` and `uint8_t b`. Then print the addition of `a` and `b` as `c` which is defined as `uint8_t c`.

- (a) What is the output when `a = 10` and `b = 100`?
- (b) Run `a = 100` `b = 250` and answer the following questions:
 - i. Why is the addition of `a` and `b` incorrect?
 - ii. 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:

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

4. More Shell Commands (1 points)

Imagine you are a space explorer on an alien planet, and you have just discovered a mysterious artifact buried in the sand. The artifact seems to have a message encoded in a strange alien language. To decipher it, you need to perform a series of bash commands.

Create a file called `alien_message.txt` and copy the following data into that file:

```
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

Find on the internet or ask ChatGPT to give you the shell commands to unveil the following hidden message:

- (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 to understand the complexity of the alien language.
- (c) Search for occurrences of the word "translate" in the file to identify potential clues.
- (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 alien word "Xylophont" with its English translation "Greetings" in the original file.

Tips You may need depending on you OS the shell commends 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 (0.5 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 (0.5 points)

Search on the internet and write a C code to create a random integer number. Is it really randomized?

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