Computer Practicum I 2022/2023

Homework 3 - C

GENERAL INSTRUCTIONS: Read carefully the steps below.

(1) There are two tasks in this homework. When done, there are two important steps needed to successfully-submit your homework.
   (a) Step 01: upload your .c files in the specified repository (Github or GitLab). **Do not ZIP them**. Refer to folder structure in item 4.
   (b) Step 02: In e-classroom, submit the github link/gitlab link of your upload to certify that your submission was successful and was made by you. (For github, submit the permalink of the file found either in your repository or in a pull request to my repository).

(2) The details of Tasks 1 and Task 2 can be found in the succeeding pages.

(3) When making submissions via git, the student should be able to show a **minimum of two commits** per submission task. Each code found in each commit **should be a compilable** version. These commits should show some form of code revision (rather than one big code submission).

(4) Your submission folder must contain two subfolders namely Task1 and Task2. The .c file must be placed in the correct subfolder and should be named **main.c**.

(5) All pull requests and link submissions must be done by 06 January 2023 23:59.

Please scroll to the next page to view Tasks 1 and 2

Task 1: Elevator (or Lift as referred to by the English folks) Simulator

Simulate an Elevator-Controller Program. The Elevator should be able to travel between five floors namely Ground, 1st, 2nd, 3rd, 4th and 5th floors. The elevator can travel going up (unless they are at the 5th floor) or going down (unless they are at the ground floor). The elevator can only contain one passenger (which is the current user of the program). Initially the elevator begins and loads at the Ground Floor. If the user wants to travel to the 5^{th} floor, the elevator shall go up to the said floor. It is the option of the user if they want to alight the elevator or not on the 5th floor or any floor on the way.

Realistic rules should apply: If the user alights at the 4^{th} floor, when they ride the elevator again, it should begin traveling from the 4^{th} floor and from there choose to go up or down. Your program should have some validation: meaning, only the valid characters should be received by the program (G, 1, 2, 3, 4, 5). If the user keys-in "A" or "-1", the program should be able to handle such input and not crash and ask for a correct valid input.

Keep your program modular (use functions), and apply all lessons and concepts learned in class. Indicate in the comments (per item) all the lessons and concepts used and if possible indicate the specific parts and syntax applied.

By completing all the features mentioned above, you earn minimum passing marks (30 points). Considerations as extra features will earn you extra points to get full marks (50 points): The elevator can also travel from B1 to B3 (Basement 1 to 3). The elevator can also allow new passengers to ride the elevator during the trip of the first user. The number of passengers per trip can vary.

Others: Do not use GOTO statements (-7 points).

Scoring system for this task (50 points):

20 points	The program performs the expected behavior according to the specs provided.
05 points	The program compiled properly.
05 points	The program can check whether the user provides the correct input or not.
10 points	Extra features were implemented and are running as expected.
05 points	The student has used the most appropriate and optimized data types.
03 points	The student has used valid and acceptable identifiers and function names.
02 points	The code is structured properly and organized in a readable way.

Task 2: Hangman game

Write a terminal based on the game of Hangman. In a hangman game, the player is given a word that they need to guess, with each letter of the word represented by an underscore/blank. The player tries to guess a letter of the word by entering it into the terminal. If the letter is correct, the blank corresponding to that letter is filled in. If the letter is incorrect, a part of a stick figure is drawn. The player has a limited number of incorrect guesses before the stick figure is fully drawn and the game is lost. The player wins the game if they guess all the letters of the word before the stick figure is fully drawn.

The program should have an array of possible words to choose from. A different word should be randomly chosen from the array every time we run the program.

Others: Do not use GOTO statements (-7 points).

Scoring system for this task (50 points):

20 points The program performs the expected behavior according to the specs provided.

05 points The program compiled properly with no warnings.

05 points The program can check whether the user provides the correct input or not.

10 points The game behaves correctly according to the rules

05 points The student has used the most appropriate and optimized data types.

05 points The code is structured properly and organized in a readable way and includes clear comments.

For questions about the homework you may contact Jordan Deja or Domen Vake.