Računalniški praktikum I 2022/2023

# Domača naloga 3 - C

SPLOŠNA NAVODILA: Pozorno preberite spodnje korake.

1. Domača naloga vsebuje 2 ločeni nalogi. Ko programirate, sta potrebna dva pomembna koraka za uspešno oddajo domače naloge.
    a. svoje datoteke .c naložite v prejektni repozitorij (Github ali GitLab). **Ne stisnite jih v ZIP.** Glejte strukturo map v točki 4.
    b. V e-učilnici oddajte povezavo do github/gitlab vašega repozitorija, da potrdite, da je bila vaša oddaja uspešna. Z oddajo izjavljate, da ste nalogo delali sami.
2. Podrobnosti 1. in 2. naloge najdete na naslednjih straneh.
3. Pri oddaji mora biti študent sposoben v git repozitoriju pokazati **najmanj** dve točki v kateri je bil projekt shranjen (**2 commita**). Vsaka koda, najdena v vsaki objavi, mora biti različica, ki jo je mogoče prevesti (**pri kompilaciji z gcc ne sme biti napak ali opozoril**). Te commiti bi morali pokazati neko obliko revizije kode (namesto ene velike predložitve kode).
4. Vaš repozitorij mora vsebovati dve podmapi, in sicer **Task1** in **Task2**. Datoteka .c mora biti postavljena v obeh podmapah in mora biti imenovana **main.c**.
5. Naloga mora biti končna in povezava predložena v e-učilnici do 6. januarja 2023 do 23:59.

**Pomaknite se na naslednjo stran, da si ogledate nalogi 1 in 2.**

# Homework 3 - C

GENERAL INSTRUCTIONS: Read carefully the steps below.

(1) There are two tasks in this homework. When programming, 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**

**Naloga 1: Simulator dvigala**

Simulirajte program za krmiljenje dvigala. Dvigalo bi moralo imeti možnost potovati med štestimi nadstropji, in sicer pritličje, 1., 2., 3., 4. in 5. nadstropje. Dvigalo se lahko dvigne (razen če je v 5. nadstropju) ali se spusti (razen če je v pritličju). Simulacija dvigala pelje enega potnika (ki je trenutni uporabnik programa). Sprva se dvigalo začne in naloži v pritličju. Če želi uporabnik potovati v 5. nadstropje, se dvigalo povzpne do omenjenega nadstropja. Izbira uporabnika je, ali želi izstopiti iz dvigala ali ne v 5. nadstropju ali katerem koli nadstropju na poti.

Veljati bi morala realistična pravila: če uporabnik izstopi v 4. nadstropju, ko se ponovno vozi z dvigalom, bi moralo začeti potovati iz 4. nadstropja in od tam izbrati, ali gre gor ali dol. Vaš program mora imeti nekaj validacije: kar pomeni, da mora program prejeti samo veljavne znake (P, 1, 2, 3, 4, 5). Če uporabnik vnese »A« ali »-1«, mora biti program sposoben obdelati tak vnos in se ne sme zaustaviti, temveč zahtevati veljaven vnos.

Naj bo vaš program modularen (uporabljajte funkcije) in uporabite vse lekcije in koncepte, pridobljene v razredu. V komentarjih (na element) navedite vse uporabljene lekcije in koncepte in, če je mogoče, navedite posebne dele in uporabljeno sintakso.

Če izpolnite vse zgoraj omenjene funkcije, si prislužite minimalne točke (30 točk). Implementcije dodatnih funkcij vam bodo prinesle polne točke (50 točk): Dvigalo lahko potuje tudi od B1 do B3 (klet 1 do 3). Dvigalo lahko omogoči tudi vožnjo novih potnikov z dvigalom med potovanjem prvega uporabnika. Število potnikov na potovanje se lahko spreminja.

**Drugo:** Ne uporabljajte stavkov GOTO (-7 točk).

**Sistem točkovanja za to nalogo (50 točk):**

| | |
|---|---|
| 20 točk | Program izvaja pričakovano obnašanje v skladu s podanimi specifikacijami. |
| 05 točk | Kompilacija programa se izvede brez opozoril in napak. |
| 05 točk | Program preveri validnost uporabniških vnosov. |
| 10 točk | Dodatne funkcije so bile implementirane in delujejo po pričakovanjih. |
| 05 točk | Študent je uporabil najustreznejše in optimizirane tipe podatkov. |
| 03 točke | Študent je uporabil veljavne in sprejemljive identifikatorje in imena funkcij. |
| 02 točki | Koda je pravilno strukturirana in organizirana na berljiv način. |

**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. |

**Naloga 2: Igra vislic**

Napišite igro na podlagi igre Hangman (vislice). V igri vislice dobi igralec besedo, ki jo mora uganiti, pri čemer je vsaka neznana črka besede predstavljena s podčrtajem. Igralec poskuša uganiti črko besede tako, da jo vnese v terminal. Če je črka pravilna, se izpolni prazen prostor, ki ustreza tej črki. Če črka ni pravilna, se nariše del slikice igralca. Igralec ima omejeno število nepravilnih poskusov, preden je slika igralca na vislicah v celoti izrisana in je igra izgubljena. Igralec zmaga v igri, če ugane vse črke besede, preden mu zmanjka poskusov.

Program mora imeti niz možnih besed, med katerimi lahko izbira. Vsakič, ko zaženemo program, je treba iz matrike naključno izbrati drugo besedo. Igra mora prav tako izpisovati vse črke, ki jih je igralec poizkusil.

Naj bo vaš program modularen (uporabljajte funkcije). V komentarjih navedite vse uporabljene lekcije in koncepte ter njihov namen.

**Drugo:** Ne uporabljajte stavkov GOTO (-7 točk).

**Sistem točkovanja za to nalogo (50 točk):**

25 točk Program izvaja pričakovano obnašanje v skladu s podanimi specifikacijami.
05 točk Kompilacija programa se izvede brez opozoril in napak.
05 točk Program preveri validnost uporabniških vnosov.
05 točk Študent je uporabil najustreznejše in optimizirane tipe podatkov.
10 točk Koda je pravilno strukturirana in organizirana na berljiv način ter vključuje jasne komentarje.

Za morebitna vprašanja o domači nalogi se lahko obrnete na asistente slovenskih ali angleških skupin.

**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. The game must also print out all the letters the player has tried.

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

**Scoring system for this task (50 points):**

25 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.
05 points The student has used the most appropriate and optimized data types.
10 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.