

Programavimo pagrindai

6 pratybos

Duomenų validacija yra PRIVALOMAS dalykas visose programose, kuriose vyksta įvedimas (nebūtinai korektiškas), nuo kurio korektiškumo priklauso ir galutinių rezultatų korektiškumas, o gal net ir tai, ar programa apskritai (ne)lūžta ir (ne)sėkmingai baigia darbą.

Šių pratybų esmė – įvaldyti C kalbos priemones validacijai vykdyti, o taip pat įgyti „nemokšoms atsparios“ (angl. foolproof) vartotojo sąsajos kūrimo įgūdžių. Kiekviena iš programų privalo pranešti: a) (pradžioje) ką ji daro b) (prieš nuskaitant) kokių tiksliai ji duomenų tikisi c) (gavus duomenis) ar duomenys sėkmingai nuskaityti ir jei ne – d) kas tiksliai buvo blogai vartotojo įvedime.

Visose programose, gavus netinkamą įvestį, turi būti prašoma kartoti, tol, kol nebus įvesta sėkmingai.

Kiekvieną užduotį rašykite atskirame kodo faile, pratybų pabaigoje visus kodo failus įkelkite į VU VMA, laikydamiesi ten pateiktų nurodymų.

Atmintinė.

- Operacijos su srautais/failais vykdomos naudojantis <stdio.h> aprašytomis funkcijomis
- Aktualios funkcijos (lmgtyf.com): fopen, fclose, feof, fgetc, fputc, fgets, fputs, fscanf, fprintf, fread, fwrite

Užduotis 1.

Parašykite programą, kuri paprašo įvesti vieną teigiamą skaičių, ir suskaičiuoja jo faktorialą, kurį atspausdina ekrane ir taip pat įrašo į rezultatų failą „out.txt“.

Užduotis 2.

Parašykite programą, kuri paprašo duomenų faile „in.txt“ pateikti realų skaičių, ne mažesnę už 10 ir nedidesnę už 1000, su ne daugiau kaip 3 skaičiais po kablelio. Laikykite, kad įvestyje trupmeninę dalį nuo sveikosios skiria kablelis. Programa turi atspausdinti įvesto skaičiaus ilgį. Jei duomenys pateikti nekorektiškai, programa turi prašyti įvesti (tikėtina, kito) duomenų failo vardą, su galimais jau korektiškais duomenimis.

Užduotis 3.

Parašykite programą, kuri paprašo įvesti tris skaičius vienoje eilutėje, be tarpų, teigiamus arba neigiamus, kuriuos vieną nuo kito skiria kabliataškiai, ir raskite vidurinį (pagal reikšmę) skaičių. Jei įvestis korektiška, rezultatą (reikšmę) programa turi atspausdinti rezultatų faile, kurio vardas įvedamas ir privalo baigtis simboliais „.txt“ (jei taip nėra, prašoma failo vardą įvesti iš naujo).

Užduotis 4.

Parašykite programą, kuri patikrina, ar eilutėje yra korektiškai įvesta data, formatu YYYY-MM-DD. Jei taip, programa turi atspausdinti, kokia tai savaitės diena. Savaitės dienų pavadinimai turi būti skaitomi iš tekstinio failo „week.txt“, kuriame yra iš anksto įrašytos septynios eilutės su atitinkamų dienų pavadinimais. Pakeitus pavadinimus faile (pvz. iš lietuvių kalbos į anglų), programoje spausdinamos reikšmės irgi turi pasikeisti (automatiškai).

Užduotis 5.

Parašykite programą, kuri patikrina, ar eilutėje yra korektiškai įvestas el.pašto adresas. Laikykite, kad jis korektiškas, jei eilutėje yra vienas simbolis @ ir (po jo) bent vienas taškas, o prieš @, po taško ir tarp šių dviejų simbolių – ne mažiau nei vienas kitoks simbolis. Jei įvestis korektiška, programa turi atspausdinti domeną, t.y. el.pašto dalį po simbolio @. Visi kada nors pateikti korektiški elektroninio pašto adresai turi būti kaupiami (append) tekstiniame faile „emails.txt“.

Papildomos užduotys.

Užduotis 6a.

Parašykite programą, kuri analizuoja faile pateiktą tekstą ir tikrina, ar jame esantys žodžiai parašyti be klaidų. Tikrinimui vykdyti, susikurkite žodyno failą, kuriame buvo kaupiami visi teisingi žodžiai. Tikrinimo metu, jei programa aptinka klaidingą (nesantį žodyne) žodį, programa turi paklausti, ar fiksuoti klaidą, ar įtraukti naują žodį į žodyną. Pirmuoju atveju, ji turi užfiksuoti (tikėtina, klaidų fiksavimui skirtame faile) klaidingą žodį ir jo poziciją (eilutės ir žodžio numerį) faile, antruoju – papildyti žodyną.

Užduotis 6b.

Realizuoti tokį duomenų saugojimo išorinėje atmintyje sprendimą, kad 6a užduotyje paminėta žodžių paieška žodyne vyktų efektyviai.

Užduotis 7.

Papildykite visas aukščiau esančias užduotis (1-5) kodo fragmentais, leidžiančiais skaičiuoti neteisingų įvedimų skaičių. Maksimalus leistinas klaidų skaičius turi būti apibrėžiamas failo viršuje su direktyva `#define`, o tokį viršijus programa turi nebe kartoti prašymą įvesti, o tiesiog baigti savo darbą.

Užduotis 8a.

Apibrėžkite funkcijų rinkinį (modulį) duomenų validacijai vykdyti, kiekvienoje iš funkcijų apibrėždami tam tikrą universalų filtrą (pvz. tikrinantį, ar reikšmė papuola į duotą intervalą; pvz. tikrinantį, ar eilutė atitinka tam tikrą formatą (arba reguliary reiškinių) ir pan.). Funkcijos turi būti tokios, kad jų pagalba galėtumėte pilnai įgyvendinti pirmų penkių užduočių sąlygas, kitaip tariant, vietoje to, kad konkreti tikrinama sąlyga būtų įgraviruota kode su fiksuotomis reikšmėmis, reikia kreiptis į atitinkamą universalią funkciją, tas fiksuotas reikšmes perduodant kaip parametrus.

Užduotis 8b.

Papildykite visas aukščiau esančias užduotis (1-5) validacijai skirta duomenų struktūra, o būtent – apibrėžkite filtrų (rodyklės į funkcijas) masyvą, ir realizuokite validacijos procesą taip, kad įvestis būtų laikoma teisinga tuomet, kai ji tenkina visus į masyvą įdėtus filtrus.