

Programavimo pagrindai

4 pratybos

Struktūrinis programavimas, kurio idėja yra programuoti apsiribojant aiškiai apibrėžtomis valdymo struktūromis (sekos, sąlygos, ciklo) ir vengti tų struktūrų neatitinkančių, chaotiškų šokinėjimų iš bet kurios kodo eilutės į bet kurią kitą (goto), stipriai palengvino kodo skaitomumą, tuo pačiu padidindamas ir kodo kokybę. Šių pratybų esmė – įvaldyti minėtas valdymo struktūras C kalboje, ir gebėti jas naudoti patiems įvairiausiems uždaviniams spręsti.

Kiekviena iš programų vartotojui turi pateikti informatyvią vartotojo sąsają, įskaitant (ir nebūtinai apsiribojant) programos pradžioje pateikiamą pranešimą ką programa daro, prašymus įvesti duomenis, paaiškinant kiek ir kokių duomenų tikimasi, patvirtinimus, jog duomenys nuskaityti sėkmingai, bei tekstą, kuriame pateikiami programos gauti rezultatai. Programa neprivalo būti apsaugota nuo nekorektiško vartotojo įvedimo, tačiau turėtų patikrinti ir atpažinti situacijas, kuomet suvedus iš pažiūros korektiškus duomenis, korektiško rezultato gauti neįmanoma. Tokiu atveju, programa turėtų atspausdinti atitinkamą pranešimą, ir gali baigti darbą. Kiekvieną užduotį rašykite atskirame kodo faile, pratybų pabaigoje visus kodo failus įkelkite į VU VMA, laikydamiesi ten pateiktų nurodymų.

Užduotis 1. Vartotojas įveda tris sveikus skaičius (a, b, c). Programa turi atspausdinti kvadratinės lygties $ax^2+bx+c=0$ sprendinių skaičių ir, jei sprendinių yra, jų reikšmes.

Užduotis 2. Vartotojas įveda tris sveikus skaičius (a, b, c). Programa turi atspausdinti visus teigiamus sveikus skaičius iš intervalo (a; b], kurie dalijasi iš skaičiaus c su liekana 1.

Užduotis 3. Vartotojas įveda tris neneigiamus sveikuosius skaičius (a, b, c). Programa turi atspausdinti c-ąjį skaičių f_c , kur $f_c=f_{c-1}+f_{c-2}$, o $f_0 = a$ ir $f_1 = b$. Jei $a=0$ ir $b=1$, f_c yra c-asis garsiosios Fibonačio sekos skaičius.

Užduotis 4. Vartotojas įveda tris natūraliuosius skaičius (a, b, c). Programa turi atspausdinti du skaičius – šių trijų skaičių didžiausią bendrąjį daliklį (DBD) bei mažiausiąjį bendrą kartotinį (MBK).

Užduotis 5. Vartotojas įveda skaičių n, o po to – n realiųjų skaičių. Programa turi suskaičiuoti šių skaičių sumą ir vidurkį, o taip pat mažiausią (minimumą) ir didžiausią (maksimumą) reikšmes.

Užduotis 6. Vartotojas įveda teigiamų skaičių seką, kurios pabaigą žymi pirma įvesta neteigiama reikšmė. Programa turi rasti, kuris skaičius tarp įvestųjų turi daugiausiai skaitmenų.

Papildomos užduotys.

Užduotis 7. Programa turi apskaičiuoti režius visiems C kalboje naudojamiems sveikiesiems duomenų tipams (*char, short, int, long, signed, unsigned* ir jų kombinacijoms). **Idėja:** prie maksimalios reikšmės pridėjus vienetą, įvyksta perpildymas (angl. overflow), ir gaunama ne vienetu didesnė, o minimali to duomenų tipo reikšmė. Norint aptikti duomenų tipo režius, galima cikle inkrementuoti atitinkamo tipo kintamąjį ir ieškoti šios neįprastos situacijos, kuriai atsitikus tampa paprasta identifikuoti minimalią ir maksimalią reikšmes.

Užduotis 8. Perrašyti pagrindinių užduočių programas, visus kode naudojamus ciklus pakeičiant a) FOR ciklais b) WHILE ciklais c) DO-WHILE ciklais d) IF+GOTO konstrukcijomis, atitinkančiomis pagrindinį kodą. **Idėja:** programavimui iš esmės pakanka vienos ciklo struktūros, ir visos kitos C kalboje yra tik patogumo dėlei; ciklo esmė yra daugkartinis vykdymas, ir trys ciklų tipai skiriasi tik sintakse ir detalėmis, kada tikrinama ciklo sąlyga ir kur ji (ir kiti veiksmi) yra užrašomi. Žinant sintaksę ir detales, nesunku vieną pakeisti kitu.