

Procedūrinio programavimo pagrindai

9 tema

Šių užduočių tikslas – susipažinti su struktūriniais duomenų tipais, išmokti jais operuoti ir kurti įvairias duomenų struktūras atmintyje.

Funkcija *main* yra skirta testavimui ir demonstracijai, kaip veikia jūsų parašytos funkcijos. Nei vienoje iš žemiau užduotyse aprašomų funkcijų negali būti nei skaitymo iš, nei spausdinimo į ekraną (nebent pačioje užduotyje nurodyta kitaip – 1b, 2c, 3c ir t.t.). Specialios vartotojo sąsajos daryti nereikia – pratybų rezultatas yra funkcijos, o *main* esantis kodas.

Kiekvieną užduotį rašykite atskirame kodo faile. Atliktas užduotis įkelkite į VU VMA, laikydamiesi pateiktų nurodymų.

Užduotis 1.

Sukurkite duomenų tipus ir funkcijas atstumui tarp dviejų taškų plokštumoje suskaičiuoti:

- apibrėžkite struktūrinį duomenų tipą *Point*, kurio viduje būtų saugomi du realūs skaičiai – taško plokštumoje koordinatės *x* ir *y*. Duomenų tipo vardui sutrumpinti pasinaudokite raktiniu žodžiu *typedef*. Funkcijoje *main* sukurkite du *Point* tipo kintamuosius *p1* ir *p2*, atitinkančius tašką su koordinatėmis (2.0, -3.0) ir tašką su koordinatėmis (-4.0, 5.0) atitinkamai
- apibrėžkite funkciją *void printPoint(Point p)*, kuri gavusi taško koordinates nusakančią struktūrą *Point*, atspausdina jo koordinates į ekraną, formatu (*x*, *y*). Pasinaudokite funkcija *printPoint* taškų *p1* ir *p2* koordinatėms atspausdinti
- apibrėžkite funkciją *Point createPoint(double x, double y)*, kuri turint du realius skaičius leistų gauti tašką su atitinkamomis koordinatėmis (sukurtų *Point* tipo struktūrą, užpildytą ją koordinatėmis, ir grąžintų tolesniam panaudojimui). Perrašykite *main* funkciją taip, kad taškai *p1* ir *p2* būtų kuriami naudojantis funkcija *createPoint*
- apibrėžkite funkciją *double getDistance(Point a, Point b)*, kuri randa (grąžina) atstumą tarp dviejų taškų plokštumoje. Perrašykite *main* funkciją taip, kad ji atliktų vieną veiksmą - apskaičiuotų atstumą tarp taškų *p1* ir *p2*. Tai reikia atlikti vienu C kalbos sakiniu – funkcijai *getDistance* tiesiogiai perduokite *createPoint* rezultata(-us), o kintamieji *p1* ir *p2* tampa nebūtini.

Užduotis 2.

Sukurkite sveikųjų skaičių steką, apibrėždami reikiamus duomenų tipus ir su jais dirbančias funkcijas:

- apibrėžkite struktūrinį duomenų tipą *Stack*, kurio viduje būtų saugomas dinaminis masyvas (rodyklė į pirmą dinaminio masyvo elementą) ir jo talpa (dydis). Duomenų tipo vardui sutrumpinti pasinaudokite žodžiu *typedef*.
- apibrėžkite funkciją *void initStack(Stack *stack)*, kuri nustatytų pradinės struktūros reikšmes (lygias 0)
- apibrėžkite funkciją *void printStack(Stack *stack)*, kuri cikle atspausdintų visus dinaminio masyvo elementus
- apibrėžkite funkciją *int getStackSize(Stack *stack)*, kuri tiesiog grąžina *Stack* viduje talpinamo dinaminio masyvo talpą (dydį)
- apibrėžkite funkciją *void push(Stack *stack, int value)*, kuri (padidinsi dinaminio masyvo talpą) įterptų naują reikšmę į pabaigą
- apibrėžkite funkciją *int top(Stack *stack)*, kuri grąžintų paskutinį dinaminio masyvo elementą (arba 0, jei masyvas tuščias).
- apibrėžkite funkciją *int pop(Stack *stack)*, kuri ne tik grąžina paskutinį dinaminio masyvo elementą (daro tą patį, ką ir funkciją *top*, ir todėl į ją kreipiasi), bet ir ištrina jį iš masyvo (atitinkamai sumažina ir dinaminio masyvo talpą)
- apibrėžkite funkciją *void destroyStack(Stack *stack)*, kuri atlaisvina visą naudojamą atmintį (atitinkamai, atnaujiną ir *Stack* viduje esančius laukus).

Daugiau informacijos: [https://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type))

Užduotis 3.

Sukurkite taškų steką, modifikuodami iki šiol sukurtus duomenų tipų aprašus ir funkcijas:

- a) modifikuokite 2 užduoties struktūrą *Stack*, nustatydami jog struktūros viduje (dinaminiam masyve) bus saugomos ne *int* tipo reikšmės, bet *Point* tipo (1 užduotis) elementai
- b) modifikuokite 2 užduoties punktuose *d-f* nurodytų funkcijų prototipus taip, kad jos dirbtų ne su *int*, o su *Point* tipo parametrais
- c) modifikuokite 2 užduotyje apibrėžtas funkcijas taip, kad jos korektiškai dirbtų su *Point* tipo duomenimis, ir pasinaudodami sukurtomis funkcijomis, į *Stack* esantį dinaminį masyvą įdėkite penkis taškus (*Point*), o tada atspausdinkite jų koordinates ir atstumus iki koordinatinių pradžios (tam modifikuokite funkciją *printStack*)

Papildomos užduotys.

Užduotis 4a.

Sukurkite vienusį sąrašą, kuriame galima saugoti sveikuosius skaičius, ir funkcijas, skirtas su juo dirbti:

- a) apibrėžkite struktūrinį duomenų tipą *List*, kurio viduje būtų saugoma *int* tipo reikšmė, ir rodyklė *next* į (sekantį vienusio sąrašo) *List* kintamąjį (elementą)
- b) apibrėžkite funkciją *List *createList(int value)*, kuri gauna sveikojo tipo reikšmę, sukuria (dinaminėje atmintyje) *List* tipo struktūrą (elementą) ir ją grąžina (rodyklė *next* nustatoma lygi *NULL*).
- c) apibrėžkite funkciją *void printList(List *list)*, kuri gauna rodyklę į *List* tipo struktūrą, atspausdina joje įrašytą sveikąją reikšmę, ir rekursyviai kreipiasi į save sekančio elemento *list→next* duomenims atspausdinti, iki tol, kol duomenys nesibaigia.
- d) apibrėžkite funkciją *int getListSize(List *list)*, kuri suskaičiuoja, kiek elementų saugoma sąrašė *List*
- d) apibrėžkite funkciją *void insertElement(List **list, int value)*, kuri gauna sveikojo tipo reikšmę, įterpia ją į sąrašo *List* pradžią
- e) apibrėžkite funkciją *int getElement(List *list)*, kuri grąžina sąrašo *List* pradžioje išsaugotą sveikojo tipo reikšmę
- f) apibrėžkite funkciją *int deleteElement(List **list)*, kuri ne tik grąžina sąrašo *List* pradžioje išsaugotą sveikojo tipo reikšmę (daro tą patį, ką ir funkcija *getElement*, todėl į ją kreipiasi), bet ir ištrina ją iš sąrašo *List* pradžios
- g) apibrėžkite funkciją *void destroyList(List **list)*, kuri sunaikina sąrašą *List*, atlaisvindama visą jam rezervuotą dinaminę atmintį (rekursijos nenaudokite)

Daugiau informacijos: https://en.wikipedia.org/wiki/Linked_list

Užduotis 4b.

Modifikuokite struktūrą *List* ir jai skirtas operacijas, kad vietoje sveikųjų skaičių ji/jos dirbtų su *Point* tipo elementais.

Užduotis 5.

Apibrėžkite dvipusį sąrašą *DoublyLinkedList*, kurio elementai yra masyvai, kurių viduje saugomos tekstinės eilutės (žodžių rinkiniai). Apibrėžkite funkcijas, leidžiančias sukurti sąrašą ir užpildyti jį duomenimis (sąrašą – masyvais, o masyvus – tekstinėmis eilutėmis). Apibrėžkite funkciją, leidžiančią surikiuoti duotą žodžių rinkinį pagal abėcėlę. Apibrėžkite funkciją, leidžiančią surikiuoti sąrašą pagal žodžių rinkinių dydį, pasirinktinai (nustatoma per funkcijos parametą) mažėjimo arba didėjimo tvarka.