

1 L4 darbas : dvimatis masyvas

1.1 Tikslai

- Išmokyti rašyti savo sukurtas C++ funkcijas su parametrais norimiems veiksams atlikti.
- Išmokyti realizuoti elementariusius skaičiavimus su statiniu dvimačiu masyvu ar atskiromis jo dalimis.
- Išmokyti pateikti programos darbo rezultatus vartotojui patogioje formoje (lentelėje), naudojant formatuotą spausdinimą: masyvas spausdinamas rėmeliuose, eilutės ir stulpeliai numeruojami.

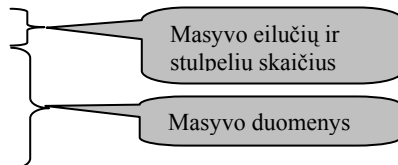
1.2 Reikalavimai

- Turi būti parašyta ir programoje naudojama bent viena C++ funkcija su parametrais.
- Pradiniai duomenys programai įvedami iš duomenų failo. Eilučių ir stulpelių skaičius nurodytas pirmoje duomenų failo eilutėje, toliau eilutėmis surašyti masyvo duomenys.
- Programos pradiniai duomenys ir darbo rezultatai spausdinami faile.

1.3 Dvimačio masyvo duomenų įvedimas iš failo

Tam, kad nereikėtų kiekvieną kartą vykdant programą pradinių duomenų įvesti klaviatūra, tikslinga rašyti programas, kurios pradinius duomenis skaito iš duomenų failo. Kadangi programa duomenis skaitys ir saugos statiniame masyve, tai uždavinio supaprastinimui galime priimti sąlygą, kad duomenų failo pirmojoje eilutėje yra parašytas dvimačio masyvo eilučių ir stulpelių skaičius, o toliau duomenų faile atskiromis eilutėmis pateikti masyvo duomenys. Duomenų failo pavyzdys ir struktūra pateikti 4.1 paveiksle.

| | | | |
|----|----|----|----|
| 3 | 4 | | |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |



4.1 pav. Duomenų failas ir jo struktūra.

Duomenų skaitymo funkcija yra tiesiogiai susijusi su duomenų failo struktūra: tik žinodami duomenų failo struktūrą, galime parašyti tokios struktūros failo skaitymo funkciją. Dvimačio masyvo duomenų įvedimo iš failo funkcijos struktūrograma pateikta 4.2 paveiksle. Norint skaityti duomenis iš failo, pirmiausia reikia atidaryti

norimą failą (tarkim "Duom.txt") skaitymui. Tada reikia perskaityti dvimačio masyvo eilučių ir stulpelių skaičių (n ir m). Žinodami, kiek masyve yra eilučių ir stulpelių, galime organizuoti ciklą cikle dvimačio masyvo duomenų skaitymui. Išorinis ciklas vyksta per masyvo eilutes: pradinė ciklo kintamojo (i) reikšmė – 0; ciklas vykdomas tol, kol ciklo kintamasis yra mažesnis už masyvo eilučių skaičių n ; atlikus ciklo veiksmus, ciklo kintamojo reikšmė didinama vienetu. Vidinis ciklas vyksta per masyvo stulpelius: pradinė ciklo kintamojo (j) reikšmė – 0; ciklas vykdomas, kol ciklo kintamasis yra mažesnis už masyvo stulpelių skaičių m ; atlikus ciklo veiksmus, ciklo kintamojo reikšmė didinama vienetu. Vidiniame funkcijos cikle perskaitomi vienos masyvo eilutės duomenys. Atlikus visus reikalingus veiksmus, būtina uždaryti duomenų failą.

| |
|--|
| Pradžia Skaityti |
| Atidaryti skaitymui srautą <code>fd(failo vardas)</code> |
| Iš <code>fd</code> skaityti n |
| Iš <code>fd</code> skaityti m |
| $i = 0, n - 1, 1$ |
| $j = 0, m - 1, 1$ |
| Iš <code>fd</code> skaityti <code>A[i][j]</code> |
| Uždaryti srautą <code>fd</code> |
| Pabaiga |

4.2 pav. Dvimačio masyvo duomenų įvedimo iš failo funkcijos struktūrograma.

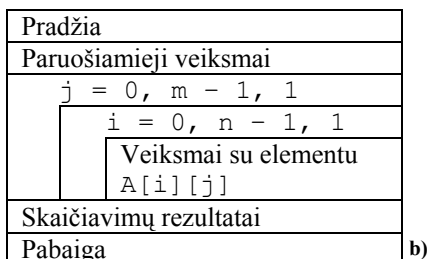
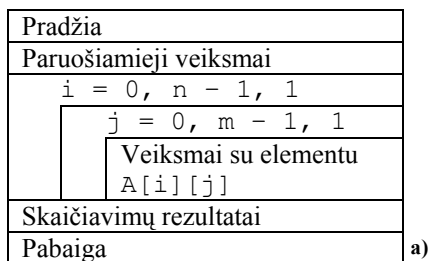
1.4 Dvimačio masyvo reikšmių spausdinimas

Norint atspausdinti dvimačio masyvo duomenis norimame faile (tarkim "Rez.txt"), pirmiausia reikia tą failą atidaryti rašymui. Jeigu to reikia, gali būti spausdinami komentarai, aprašantys toliau faile pateikiamą informaciją. Dvimačio masyvo duomenų spausdinimui faile vykdomas ciklas cikle: išorinis ciklas vyksta per masyvo eilutes, o vidinis – per masyvo stulpelius. Vidinis funkcijos ciklas atspausdina vienos masyvo eilutės duomenis. Atlikus visus reikalingus veiksmus, būtina uždaryti rezultatų failą. Dvimačio masyvo duomenų spausdinimo faile funkcijos struktūrograma pateikta 4.3 paveiksle.

| |
|--|
| Pradžia Spausdinti |
| Atidaryti rašymui srautą <code>fr(failo vardas)</code> |
| I <code>fr</code> spausdinti komentarus |
| $i = 0, n - 1, 1$ |
| $j = 0, m - 1, 1$ |
| I <code>fr</code> spausdinti <code>A[i][j]</code> |
| Uždaryti srautą <code>fr</code> |
| Pabaiga |

1.5 Veiksmai su visomis masyvo reikšmėmis

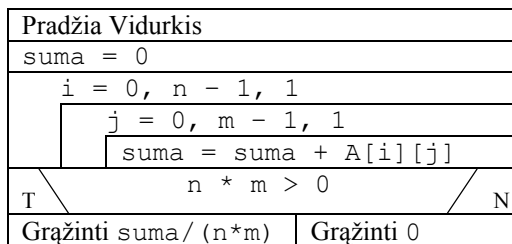
Veiksmai su visomis dvimačio masyvo reikšmėmis atliekami pagal vieną iš 4.4 paveiksle pateiktų šablonų. Abiem atvejais prieš ciklą cikle yra atliekami parengiamieji veiksmai: dažniausiai tai yra pradinių reikšmių suteikimas skaičiavimams reikalingiems kintamiesiems. Ciklas cikle gali būti organizuojamas dviem būdais. Pirmuoju atveju (4.4.a pav.) išorinis ciklas vyksta per masyvo eilutes, o vidinis – per masyvo stulpelius: vidiniame cikle analizuojami vienos masyvo eilutės duomenys. Antruoju atveju (4.4.b pav.) išorinis ciklas vyksta per masyvo stulpelius, o vidinis – per masyvo eilutes: vidiniame cikle analizuojami vieno masyvo stulpelio duomenys. Kurį variantą reiktų pasirinkti, priklauso nuo sprendžiamo uždavinio.



4.4 pav. Dvimačio masyvo duomenų nagrinėjimo eilutėmis (a) arba stulpeliais (b) struktūrogramos.

1.5.1 Masyvo reikšmių vidurkis

Dvimačio masyvo reikšmių vidurkio skaičiavimo funkcijos algoritmas pateiktas 4.5 paveiksle. Parengiamųjų veiksmų dalyje kintamajam suma suteikiama pradinė reikšmė 0: pradžioje turime sukaupę nulį. Toliau vykdomas ciklas cikle: vidiniame cikle prie turimos sumos pridedame eilinę masyvo elemento reikšmę. Pasibaigus ciklams, funkcija gali grąžinti gautą rezultatą. Čia reikia įvertinti tą situaciją, kai masyve nėra nė vienos reikšmės, kadangi tokiu atveju vidurkio suskaičiuoti negalima (dalyba iš 0 negalima).



| |
|---------|
| Pabaiga |
|---------|

4.5 pav. Dvimačio masyvo reikšmių sumos skaičiavimo funkcijos struktūrograma.

1.5.2 Teigiamų masyvo reikšmių kiekis

Dvimačio masyvo teigiamų reikšmių kiekio skaičiavimo funkcijos algoritmas pateiktas 4.6 paveiksle. Parengiamųjų veiksmų dalyje kintamajam `kiekis` suteikiama pradinė reikšmė 0: pradžioje neturim nė vieno teigiamo skaičiaus. Toliau vykdomas ciklas cikle. Vidiniame cikle analizuojama eilinė masyvo elemento reikšmė: jei ji yra didesnė už 0 (t.y. teigiamas skaičius), tai reiškia, kad dabar turime viena teigiama reikšmę daugiau – kintamojo `kiekis` reikšmę reikia padidinti vienetu. Pasibaigus ciklams, funkcija gali grąžinti gautą rezultatą – kintamojo `kiekis` reikšmę.

| | |
|---------------------|-------------|
| Pradžia TeigKiekis | |
| kiekis = 0 | |
| i = 0, n - 1, 1 | |
| j = 0, m - 1, 1 | |
| T | A[i][j] > 0 |
| kiekis = kiekis + 1 | |
| N | |
| Grąžinti kiekis | |
| Pabaiga | |

4.6 pav. Dvimačio masyvo teigiamų reikšmių kiekio skaičiavimo funkcijos struktūrograma.

1.5.3 Didžiausia masyvo reikšmė ir jos vieta

Dvimačio masyvo didžiausios reikšmės ir jos vietos masyve radimo funkcijos algoritmas pateiktas 4.7 paveiksle. Reikšmės vietą dvimačiame masyve nusako 2 skaičiai: eilutės numeris ir stulpelio numeris. Pradžioje besąlygiškai priimkime, kad didžiausia masyvo reikšmė yra pati pirmoji (masyvo narys su eilutės numeriu 0 ir stulpelio numeriu 0). Todėl parengiamųjų veiksmų dalyje kintamieji `di` ir `dj` įgyja pradinę reikšmę 0, o kintamasis `did` – `A[di][dj]`. Toliau vykdomas ciklas cikle. Vidiniame cikle analizuojama eilinė masyvo elemento reikšmė. Jei ji yra didesnė už šiuo metu žinomą didžiausią reikšmę (kintamąjį `did`), vadinasi ją reikia įsiminti kaip didžiausią: kintamajam `di` įrašomas dabar nagrinėjamos eilutės numeris, kintamajam `dj` – dabar nagrinėjamo stulpelio numeris, o kintamajam `did` – dabar nagrinėjama masyvo reikšmė. Pasibaigus ciklams, kintamuosiuose `di`, `dj` ir `did` turime skaičiavimų rezultatus.

| |
|-----------------------|
| Pradžia RastiDidVieta |
| di = dj = 0 |
| did = A[di][dj] |

| | |
|-----------------|---------------|
| i = 0, n - 1, 1 | |
| j = 0, m - 1, 1 | |
| T | A[i][j] > did |
| N | |
| di = i | |
| dj = j | |
| did = A[di][dj] | |
| Pabaiga | |

4.7 pav. Dvimačio masyvo didžiausios reikšmės ir jos vietos masyve radimo funkcijos struktūrograma.

1.6 Veiksmai su masyvo eilutės reikšmėmis

Atliekant veiksmus su masyvo eilučių reikšmėmis, parengiamoji skaičiavimų dalis persikelia į išorinį ciklą: prieš pradedant veiksmus eilutėje reikia tam pasiruošti. Vidiniame cikle atliekama tos masyvo eilutės reikšmių analizė. Pasibaigus vidiniam ciklui, jau turime veiksmų su eilute rezultatą ir galime su juo atlikti reikiamus veiksmus. Veiksmų su masyvo eilučių reikšmėmis šablonas pateiktas 4.8 paveiksle.

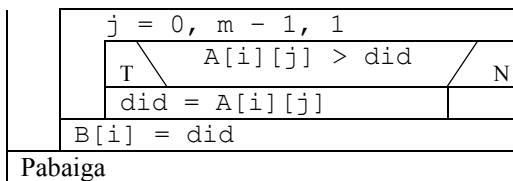
| | |
|-----------------------------|--|
| Pradžia | |
| i = 0, n - 1, 1 | |
| Paruošiamieji veiksmai | |
| j = 0, m - 1, 1 | |
| Veiksmai su A[i][j] reikšme | |
| Skaičiavimų rezultatai | |
| Pabaiga | |

4.8 pav. Dvimačio masyvo eilučių reikšmių nagrinėjimo struktūrograma.

1.6.1 Didžiausių reikšmių eilutėse radimas

Išnagrinėjame situaciją, kai reikia rasti kiekvienos eilutės didžiausią reikšmę, o surastas reikšmes reikia įrašyti į vienmatį masyvą B. Šiuo atveju vienmačiame masyve bus tiek reikšmių, kiek dvimačiame masyve yra eilučių. Parengiamųjų skaičiavimų dalyje kintamasis did įgyja savo pradinę reikšmę: pradžioje besąlygiškai priimam, kad šiuo metu žinoma didžiausia reikšmė yra pats pirmasis elementas toje eilutėje. Vidiniame cikle ieškoma didžiausia eilutės reikšmė: jeigu nagrinėjamo masyvo eilutės elemento reikšmė yra didesnė už šiuo metu žinomą didžiausią reikšmę, vadinasi ją reikia įrašyti į kintamąjį did. Pasibaigus vidiniam ciklui, jau turime surastą didžiausią reikšmę eilutėje ir galime ją įrašyti į atitinkamą masyvo B elementą. Šios funkcijos struktūrograma pateikta 4.9 paveiksle.

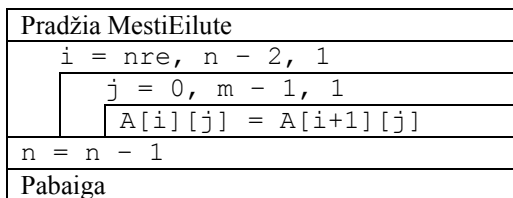
| | |
|-----------------------------|--|
| Pradžia RastiEilDidReiksmes | |
| i = 0, n - 1, 1 | |
| did = A[i][0] | |



4.9 pav. Dvimačio masyvo eilučių didžiausių reikšmių radimo funkcijos struktūrograma.

1.6.2 Eilutės pašalinimas

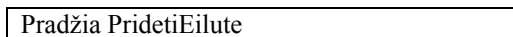
Funkcijos, pašalinančios iš dvimačio masyvo nurodytą eilutę (kintamojo *nre* reikšmė), struktūrograma pateikta 4.10 paveiksle. Funkcijos veikimo idėja paprasta: visos masyvo eilutės, esančios toliau eilutės su nurodytu numeriu (*nre*), turi būti perkeltos viena eilute arčiau masyvo pradžios. Todėl išorinio ciklo kintamasis įgyja pradinę reikšmę *nre*. Išorinis ciklas vykdomas vienu žingsniu mažiau, kadangi vidiniame cikle į atitinkamą poziciją įrašomi tolimesnės eilutės duomenys. Pasibaigus vidiniam ciklui, masyve turėsime vieną eilutę mažiau, todėl reikia nepamiršti masyvo eilučių kiekį saugančio kintamojo (*n*) reikšmę sumažinti 1.



4.10 pav. Dvimačio masyvo nurodytos eilutės pašalinimo funkcijos struktūrograma.

1.6.3 Eilutės įterpimas

Funkcijos, įterpiančios į dvimačio masyvo nurodytą (kintamojo *nre* reikšmė) eilutę kito masyvo (vienmačio masyvo *B*) reikšmes, struktūrograma pateikta 4.11 paveiksle. Vienmačiame masyve turi būti tiek reikšmių, kiek dvimačiame masyve yra stulpelių. Iš esmės funkcija veikia priešingai nei eilutės pašalinimo funkcija. Čia visos masyvo eilutės, pradedant eilute su nurodytu numeriu (*nre*), turi būti perkeltos viena eilute toliau nuo masyvo pradžios. Todėl išorinis ciklas prasideda nuo paskutiniosios masyvo eilutės numerio (*n-1*) ir mažėja, kol pasiekia nurodytą reikšmę (*nre*). Vidiniame cikle į tolimesnę eilutę perkeliame nagrinėjamos eilutės duomenys. Pasibaigus vidiniam ciklui, masyve turėsime vieną eilutę daugiau. Dabar galima į norimą vietą (*nre*) masyve įrašyti masyvo *B* reikšmes. Tą atlieka atskiras ciklas. Po reikšmių įterpimo reikia nepamiršti masyvo eilučių kiekį saugančio kintamojo (*n*) reikšmę padidinti 1.



| |
|---------------------|
| i = n - 1, nre, -1 |
| j = 0, m - 1, 1 |
| A[i+1][j] = A[i][j] |
| j = 0, m - 1, 1 |
| A[nre][j] = B[j] |
| n = n + 1 |
| Pabaiga |

4.11 pav. Naujos eilutės įterpimo į dvimatį masyvą funkcijos struktūrograma.

1.7 Veiksmai su masyvo stulpelių reikšmėmis

Atliekant veiksmus su masyvo stulpelių reikšmėmis, vidinis ir išorinis ciklai susikeičia vietomis: masyvas turi būti nagrinėjamas ne eilutėmis, bet stulpeliais. Parengiamoji skaičiavimų dalis turi būti išoriniame cikle: prieš pradedant veiksmus stulpelyje reikia tam pasiruošti. Vidiniame cikle atliekama to masyvo stulpelio reikšmių analizė. Pasibaigus vidiniam ciklui, jau turime veiksmų su stulpeliu rezultatą ir galime su juo atlikti reikiamus veiksmus. Veiksmų su masyvo stulpelių reikšmėmis šablonas pateiktas 4.12 paveiksle.

| |
|-----------------------------|
| Pradžia |
| j = 0, m - 1, 1 |
| Paruošiamieji veiksmai |
| i = 0, n - 1, 1 |
| Veiksmai su A[i][j] reikšme |
| Skaičiavimų rezultatai |
| Pabaiga |

4.12 pav. Dvimačio masyvo stulpelių reikšmių nagrinėjimo struktūrograma.

1.7.1 Stulpelių reikšmių sumų radimas

Išnagrinėjame situaciją, kai reikia rasti kiekvieno stulpelio reikšmių sumą, o surastas reikšmės reikia įrašyti į vienmatį masyvą B. Vienmačiame masyve bus tiek reikšmių, kiek dvimačiame masyve yra stulpelių. Parengiamųjų skaičiavimų dalyje kintamasis suma įgyja savo pradinę reikšmę 0. Vidiniame cikle skaičiuojama nagrinėjamo masyvo stulpelio reikšmių suma: prie kintamojo suma saugomos reikšmės pridedama eilinė masyvo elemento reikšmė. Pasibaigus vidiniam ciklui, jau turime surastą stulpelio reikšmių sumą ir galime ją įrašyti į atitinkamą masyvo B elementą. Šios funkcijos struktūrograma pateikta 4.13 paveiksle.

| |
|------------------------------|
| Pradžia RastiStulpReiksSumas |
| j = 0, m - 1, 1 |
| suma = 0 |

| | |
|---------|-------------------------|
| | $i = 0, n - 1, 1$ |
| | $suma = suma + A[i][j]$ |
| | $B[j] = suma$ |
| Pabaiga | |

4.13 pav. Dvimačio masyvo stulpelių reikšmių sumų radimo funkcijos struktūrograma.

1.7.2 Stulpelio pašalinimas

Funkcijos, pašalinančios iš dvimačio masyvo nurodytą stulpelį (kintamojo *nrs* reikšmė), struktūrograma pateikta 4.14 paveiksle. Funkcijos veikimo idėja analogiška prieš tai pateiktai eilutės šalinimo funkcijai: visi masyvo stulpeliai, esantys toliau stulpelio su nurodytu numeriu (*nrs*), turi būti perkelti vienu stulpeliu arčiau masyvo pradžios. Čia masyvas nagrinėjamas stulpeliais, t.y. išorinis ciklas eina per stulpelius, o vidinis – per eilutes. Išorinio ciklo kintamasis įgyja pradinę reikšmę *nrs*. Išorinis ciklas vykdomas vienu žingsniu mažiau, kadangi vidiniame cikle į atitinkamą poziciją įrašomi tolimesnio stulpelio duomenys. Pasibaigus vidiniam ciklui, masyve turėsime vienu stulpeliu mažiau, todėl reikia nepamiršti masyvo stulpelių kiekį saugančio kintamojo (*m*) reikšmę sumažinti 1.

| | |
|-----------------------|-----------------------|
| Pradžia MestiStulpeli | |
| | $j = nrs, m - 2, 1$ |
| | $i = 0, n - 1, 1$ |
| | $A[i][j] = A[i][j+1]$ |
| $m = m - 1$ | |
| Pabaiga | |

4.14 pav. Dvimačio masyvo nurodyto stulpelio pašalinimo funkcijos struktūrograma.

1.7.3 Stulpelio įterpimas

Funkcijos, įterpiančios į dvimačio masyvo nurodytą (kintamojo *nrs* reikšmė) stulpelį kito masyvo (vienmačio masyvo *B*) reikšmes, struktūrograma pateikta 4.15 paveiksle. Vienmačiame masyve turi būti tiek reikšmių, kiek dvimačiame masyve yra eilučių. Iš esmės funkcija veikia priešingai nei stulpelio pašalinimo funkcija. Čia visi masyvo stulpeliai, pradedant stulpeliu su nurodytu numeriu (*nrs*), turi būti perkelti vienu stulpeliu toliau nuo masyvo pradžios. Išorinis ciklas prasideda nuo paskutiniojo masyvo stulpelio numerio (*m-1*) ir mažėja, kol pasiekia nurodytą reikšmę (*nrs*). Vidiniame cikle į tolimesnį stulpelį perkeliame nagrinėjamo stulpelio duomenys. Pasibaigus vidiniam ciklui, masyve turėsime vienu stulpeliu daugiau. Dabar galima į norimą vietą (*nrs*) masyve įrašyti masyvo *B* reikšmes. Tą atlieka atskiras ciklas. Po reikšmių įterpimo reikia nepamiršti masyvo stulpelių kiekį saugančio kintamojo (*m*) reikšmę padidinti 1.

| | | | | |
|--|--------------------|--|-----------------|---------------------|
| Pradžią PridėtiStulpeli | | | | |
| <table> <tr> <td>j = m - 1, nrs, -1</td></tr> <tr> <td> <table> <tr> <td>j = 0, n - 1, 1</td></tr> <tr> <td>A[i][j+1] = A[i][j]</td></tr> </table> </td></tr> </table> | j = m - 1, nrs, -1 | <table> <tr> <td>j = 0, n - 1, 1</td></tr> <tr> <td>A[i][j+1] = A[i][j]</td></tr> </table> | j = 0, n - 1, 1 | A[i][j+1] = A[i][j] |
| j = m - 1, nrs, -1 | | | | |
| <table> <tr> <td>j = 0, n - 1, 1</td></tr> <tr> <td>A[i][j+1] = A[i][j]</td></tr> </table> | j = 0, n - 1, 1 | A[i][j+1] = A[i][j] | | |
| j = 0, n - 1, 1 | | | | |
| A[i][j+1] = A[i][j] | | | | |
| i = 0, n - 1, 1 | | | | |
| A[i][nrs] = B[i] | | | | |
| M = m + 1 | | | | |
| Pabaiga | | | | |

4.15 pav. Naujo stulpelio įterpimo į dvimatį masyvą funkcijos struktūrograma.

1.8 Veiksmai su masyvo dalimis

Atliekant veiksmus su tam tikromis dvimačio masyvo dalimis (pusėmis, ketvičiais ir pan.), viskas priklauso nuo masyvo reikšmių indeksų. Reikia tik teisingai užrašyti, kaip cikluose turi kisti ciklo kintamųjų (masyvo reikšmių indeksų) reikšmės ir cikluose bus nagrinėjama kita norima masyvo dalis. Keletas masyvo dalių nagrinėjimo šablonų fragmentų yra pateikta 4.16 paveiksle.

| | |
|---------------------------------|---------------------------------|
| i = n/2 + 1, n - 1, 1 | i = n/2 + 1, n - 1, 1 |
| j = 0, m - 1, 1 | j = 0, m/2, 1 |
| Veiksmai su elementu A[i][j] | Veiksmai su elementu A[i][j] |

4.16 pav. Veiksmų su dvimačio masyvo dalimis struktūrogramų fragmentai.

1.9 Uždavinių pavyzdžiai

1.9.1 1 uždavinys

1.9.1.1 Užduotis

Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą A(n, m). Kiekvienoje eilutėje vietoje pirmo skaičiaus užrašyti tos eilutės teigiamų skaičių sumą, o vietoje paskutinio skaičiaus – eilutės mažiausią skaičių. Faile spausdinti masyvą A(n, m) prieš ir po keitimų.

1.9.1.2 Programos tekstas

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
```

```

using namespace std;

// Konstantos
const int Cn = 100;
const int Cm = 50;
const string CDuom = "Masyvas.txt";
const string CRez = "Rezultatai.txt";

// Funkcijų prototipai
void Skaityti( const string failas, int Duom[][Cm],
               int &n, int &m);
void Linija( ofstream &fr, int k);
void Antraste( ofstream &fr, string tekstas, int m);
void Rasyti( const string failas, string tekstas,
             int Duom[][Cm], int n, int m);
void Formuoti( int Duom[][Cm], int n, int m);
int Suma( int Mas[], int k);
int Mini( int Mas[], int k);

int main ()
{
    int Mas[Cn][Cm];
    int n, m;

    ofstream fr( CRez.c_str());
    fr.close();

    Skaityti( CDuom, Mas, n, m);
    Rasyti( CRez, " Pradiniai duomenys", Mas, n, m);

    Formuoti( Mas, n, m);
    Rasyti( CRez, " Skaičiavimų rezultatai", Mas, n, m);

    cout << "\nRezultatų ieškokite faile " << CRez.c_str()
    << endl;

    return 0;
}

// Įveda duomenis iš failo į dvimatį masyvą.
// failas – duomenų failo vardas
// Duom – dvimatis masyvas
// n – masyvo eilučių kiekis

```

```
// m – masyvo stulpelių kiekis
void Skaityti( const string failas, int Duom[][Cm],
               int &n, int &m)
{
    ifstream fd( failas.c_str());

    fd >> n >> m;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            fd >> Duom[i][j];

    fd.close();
}
```

```
// Spausdina nurodyto ilgio liniją į srautą.
// fr – srautas, į kurį spausdinama
// k – spausdinamos linijos ilgis
void Linija( ofstream &fr, int k)
{
    fr << " +";
    for (int i = 0; i < k; i++)
        fr << "-----+";
    fr << endl;
}
```

```
// Spausdina lentelės antraštę į tekstinį failą.
// fr – srautas, į kurį spausdinama
// tekstas – aiškinamasis tekstas, spausdinamas prieš lentelę
// m – masyvo stulpelių kiekis
void Antraste( ofstream &fr, string tekstas, int m)
{
    fr << endl << tekstas.c_str() << endl;
    Linija( fr, m+1);
    fr << " | E.\\S. | ";
    for (int j = 0; j < m; j++)
        fr << setw( 5) << j+1 << " | ";
    fr << endl;
    Linija( fr, m+1);
}
```

```
// Spausdina masyvo duomenis į tekstinį failą.
// failas – duomenų failo vardas
// tekstas – aiškinamasis tekstas, spausdinamas prieš lentelę
```

```

// Duom – dvimatis masyvas
// n – masyvo eilučių kiekis
// m – masyvo stulpelių kiekis
void Rasyti( const string failas, string tekstas,
             int Duom[][Cm], int n, int m)
{
    ofstream fr( failas.c_str(), ios::app);
    Antraste( fr, tekstas, m);

    for (int i = 0; i < n; i++){
        fr << " | " << setw( 5) << i+1 << " | ";
        for (int j = 0; j < m; j++)
            fr << setw( 5) << Duom[i][j] << " | ";
        fr << endl;
    }

    Linija( fr, m+1);
    fr << endl;

    fr.close();
}

// Randa ir grąžina vienmačio masyvo teigiamų elementų sumą.
// Mas – vienmatis masyvas
// k – masyvo elementų kiekis
int Suma( int Mas[], int k)
{
    int s = 0;
    for (int i = 0; i < k; i++)
        if (Mas[i] > 0 )
            s = s + Mas[i];
    return s;
}

// Randa ir grąžina vienmačio masyvo mažiausio elemento reikšmę.
// Mas – vienmatis masyvas
// k – masyvo elementų kiekis
int Mini( int Mas[], int k)
{
    int mini = Mas[0];
    for (int i = 1; i < k; i++)
        if (Mas[i] < mini)
            mini = Mas[i];
}

```

```

    return mini;
}

// Kiekvienoje eilutėje vietoje pirmo skaičiaus įrašo eilutės teigiamų skaičių sumą,
// o vietoje paskutinio skaičiaus – eilutės mažiausią skaičių
// Duom – dvimatis masyvas
// n – masyvo eilučių kiekis
// m – masyvo stulpelių kiekis
void Formuoti( int Duom[][Cm], int n, int m)
{
    int sum, min;
    for (int i = 0; i < n; i++){
        sum = Suma( Duom[i], m);
        min = Mini( Duom[i], m);
        Duom[i][0] = sum;
        Duom[i][m-1] = min;
    }
}

```

1.9.2 2 uždavinys

1.9.2.1 Užduotis

Duoti duomenys apie knygynus. Turime n knygynų ir m pavadinimų knygų. Masyve Knygos(n , m) yra surašyti knygynuose turimų knygų egzempliorių kiekiai. Masyve Parduotos(n , m) yra surašyti knygynuose per dieną parduotų knygų egzempliorių kiekiai. Suformuokite daugiausiai kiekviename knygyne likusių knygų lentelę: knygyno numeris, knygos numeris, likusių knygų egzempliorių kiekis.

1.9.2.2 Programos tekstas

```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
using namespace std;

// Konstantos
const int Cn = 100;
const int Cm = 20;
const string CDuomKnygos = "Knygos.txt";
const string CDuomParduotos = "Parduotos.txt";
const string CRez = "Rezultatai.txt";

```

```

// Funkcijų prototipai
void Skaityti( const string failas, int Duom[][Cm],
               int &n, int &m);
void Rasyti( const string failas, string tekstas,
              int Duom[][Cm], int n, int m);
void Likutis( int Turim[][Cm], int Parduota[][Cm],
               int n, int m);
void Formuoti( int Duom[][Cm], int Daug[][Cm],
                int n, int m);
int Maxi( int Mas[], int k);

int main ()
{
    int Knygos[Cn][Cm];           // visos knygos
    int Parduotos[Cn][Cm];        // parduotos knygos
    int Daugiausia[Cn][Cm];       // daugiausia turima knygų
    int n, m;                     // masyvų eilučių ir stulpelių kiekiai

    ofstream fr( CRez.c_str());
    fr.close();

    Skaityti( CDuomKnygos, Knygos, n, m);
    Rasyti( CRez, "Knygynuose turimų knygų kiekiai",
            Knygos, n, m);
    Skaityti( CDuomParduotos, Parduotos, n, m);
    Rasyti( CRez, "Knygynuose parduotų knygų kiekiai",
            Parduotos, n, m);

    Likutis( Knygos, Parduotos, n, m);
    Formuoti( Knygos, Daugiausia, n, m);
    Rasyti( CRez, "Knygynuose likusių knygų kiekiai",
            Knygos, n, m);
    Rasyti( CRez, "Knygynuose daugiausia likusių knygų
                kiekiai", Daugiausia, n, 3);

    cout << "\nRezultatų ieškokite faile " << CRez.c_str()
    << endl;

    return 0;
}

// Įveda duomenis iš failo į dvimatį masyvą.

```

```

// failas – duomenų failo vardas
// Duom – dvimatis masyvas
// n – masyvo eilučių kiekis
// m – masyvo stulpelių kiekis
void Skaityti( const string failas, int Duom[][Cm],
               int &n, int &m)
{
    ifstream fd( failas.c_str());

    fd >> n >> m;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            fd >> Duom[i][j];

    fd.close();
}

// Spausdina nurodyto ilgio liniją į srautą.
// fr – srautas, į kurį spausdinama
// k – spausdinamos linijos ilgis
void Linija( ofstream &fr, int k)
{
    fr << " +";
    for (int i = 0; i < k; i++)
        fr << "-----+";
    fr << endl;
}

// Spausdina lentelės antraštę į tekstinį failą.
// fr – srautas, į kurį spausdinama
// tekstas – aiškinamasis tekstas, spausdinamas prieš lentelę
// m – masyvo stulpelių kiekis
void Antraste( ofstream &fr, string tekstas, int m)
{
    fr << endl << tekstas.c_str() << endl;
    Linija( fr, m+1);
    fr << " | E.\\S. | ";
    for (int j = 0; j < m; j++)
        fr << setw( 5) << j+1 << " | ";
    fr << endl;
    Linija( fr, m+1);
}

```

```

// Spausdina masyvo duomenis į tekstinį failą.
// failas – duomenų failo vardas
// tekstas – aiškinamasis tekstas, spausdinamas prieš lentelę
// Duom – dvimatis masyvas
// n – masyvo eilučių kiekis
// m – masyvo stulpelių kiekis
void Rasyti( const string failas, string tekstas,
              int Duom[][Cm], int n, int m)
{
    ofstream fr( failas.c_str(), ios::app);
    Antraste( fr, tekstas, m);

    for (int i = 0; i < n; i++){
        fr << " | " << setw( 5) << i+1 << " | ";
        for (int j = 0; j < m; j++)
            fr << setw( 5) << Duom[i][j] << " | ";
        fr << endl;
    }

    Linija( fr, m+1);
    fr << endl;

    fr.close();
}

// Suskaičiuoja knygų likutį (atleika dvimačių masyvų atimtį).
// Turim – dvimatis masyvas (iš kurio atimama)
// Parduota – dvimatis masyvas (kurį atima)
// n – masyvų eilučių kiekis
// m – masyvų stulpelių kiekis
void Likutis( int Turim[][Cm], int Parduota[][Cm],
              int n, int m)
{
    for( int i = 0; i < n; i++)
        for( int j = 0; j < m; j++)
            Turim[i][j] = Turim[i][j] - Parduota[i][j];
}

// Randa ir grąžina vienmačio masyvo didžiausio elemento indeksą.
// Mas – vienmatis masyvas
// k – masyvo elementų kiekis
int Maxi( int Mas[], int k)
{

```

```

int maxi = 0;
for (int i = 1; i < k; i++)
    if (Mas[i] > Mas[maxi])
        maxi = i;
return maxi;
}

```

// Suformuoja rezultatų masyvą (kurių knygų liko daugiausia kiekviename knygyne).

// Duom – dvimatis masyvas (duomenys)

// Daug – dvimatis masyvas (suformuotas rezultatas)

// n – masyvų eilučių kiekis

// m – masyvų stulpelių kiekis

```

void Formuoti( int Duom[][Cm], int Daug[][Cm],
               int n, int m)
{
    for (int i = 0; i < n; i++){
        Daug[i][0] = i + 1;
        Daug[i][1] = Maxi( Duom[i], m) + 1;
        Daug[i][2] = Duom[i][Daug[i][1] - 1];
    }
}

```

1.10 Savikontrolės klausimai ir užduotys

- Papildykite savo programą funkcija, kuri suskaičiuotų nurodyto stulpelio elementų su lyginiais eilutės numeriais sumą. Stulpelio numeris įvedamas klaviatūra ir perduodamas funkcijai per parametrų sąrašą.
- Papildykite savo programą funkcija, kuri suskaičiuotų, kiek visame masyve yra elementų, kurie be liekanos dalinasi iš skaičiaus 3.
- Papildykite savo programą funkcija, kuri rastų masyvo trečiojo ketvirčio didžiausią elementą.
- Papildykite savo programą funkcija, kuri suskaičiuotų, kiek masyve yra elementų, didesnių už elementą, esantį masyvo viduryje.
- Papildykite savo programą funkcija, kuri paskutiniojo masyvo stulpelio elementų sumą įrašytų į patį pirmąjį masyvo elementą.

1.11 Individualios L4 darbo užduotys

1. Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekvienoje eilutėje vietoje pirmo skaičiaus užrašyti tos eilutės

skaičių sumą, o vietoje paskutinio skaičiaus – eilutės didžiausią skaičių. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų.

2. Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekviename stulpelyje vietoje pirmo skaičiaus užrašyti to stulpelio teigiamų skaičių sumą, o vietoje paskutinio skaičiaus – stulpelio didžiausią skaičių. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų.

3. Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekvienoje eilutėje vietoje pirmo skaičiaus užrašyti tos eilutės teigiamų skaičių kiekį, o vietoje paskutinio skaičiaus – neigiamų skaičių kiekį. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų.

4. Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekvienoje eilutėje vietoje pirmo skaičiaus užrašyti tos eilutės neigiamų skaičių sumą, o vietoje paskutinio skaičiaus – eilutės mažiausią skaičių. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų.

5. Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekviename stulpelyje vietoje pirmo skaičiaus užrašyti to stulpelio teigiamų skaičių kiekį, o vietoje paskutinio skaičiaus – neigiamų skaičių kiekį. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų.

6. Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekviename stulpelyje rasti stulpelio elementų vidurkio sveikąją dalį ir visus stulpelio narius, kurie dalijasi iš jo be liekanos, pakeisti -1 . Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų.

7. Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekvienoje eilutėje rasti, kiek yra narių, kurie didesni už savo kaimynus. Rastus skaičius surašyti į masyvą $B(n)$. Masyvą $B(n)$ įterpti į masyvo $A(n, m)$ vidurį. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų bei masyvą $B(n)$.

8. Tekstiniame duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekviename stulpelyje rasti, kiek yra narių, kurie mažesni už savo kaimynus. Rastus skaičius surašyti į masyvą $B(m)$. Masyvą $B(m)$ įterpti į masyvo $A(n, m)$ vidurį. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų bei masyvą $B(m)$.

9. Tekstiniam duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekvienoje eilutėje rasti, kiek yra narių, kurie didesni už savo kaimynų sumą. Rastus skaičius surašyti į masyvą $B(n)$. Masyvą $B(n)$ įterpti į masyvo $A(n, m)$ pradžią. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų bei masyvą $B(n)$.

10. Tekstiniam duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, m)$. Kiekviename stulpelyje rasti, kiek yra narių, kurie mažesni už savo kaimynų sumą. Rastus skaičius surašyti į masyvą $B(m)$. Masyvą $B(m)$ įterpti į masyvo $A(n, m)$ pradžią. Faile spausdinti masyvą $A(n, m)$ prieš ir po keitimų bei masyvą $B(m)$.

11. Tekstiniam duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, n)$. Rasti masyvo narių, esančių virš pagrindinės įstrižainės, sumą. Masyvo narius, esančius žemiau pagrindinės įstrižainės ir didesnius už surastą sumą, surašyti į masyvą $B(k)$. Faile spausdinti masyvą $A(n, n)$ bei masyvą $B(k)$.

12. Tekstiniam duomenų faile pirmoje eilutėje užrašyti dvimačio masyvo eilučių ir stulpelių skaičiai. Kitose eilutėse yra masyvo elementų reikšmės. Reikia įvesti skaičius į masyvą $A(n, n)$. Rasti masyvo narių, esančių žemiau šalutinės įstrižainės, sandaugą. Masyvo narius, esančius virš šalutinės įstrižainės ir mažesnius už surastą sandaugą, surašyti į masyvą $B(k)$. Faile spausdinti masyvą $A(n, n)$ bei masyvą $B(k)$.