

Masyvai

Informacinės technologijos

22 - 23 pamokos uždaviniai

- Susipažinti su masyvu ir jo aprašymu;
- Išmokti skaityti duomenis iš failo į masyvą, kai žinomas duomenų kiekis;
- Išmokti masyvo reikšmes rašyti į failą lentele;
- Pritaikyti sumos, kiekio ir vidurkio skaičiavimo algoritmus darbui su masyvo reikšmėmis.

Masyvo duomenų struktūra

Masīvas – viena pagrindinių duomenų struktūrų, naudojamų programavimo kalbose. Ši struktūra leidžia laikyti atmintyje (vienoje vietoje) vienu vardu daug to paties tipo reikšmių. Pavyzdžiui, žinomos vieno mėnesio pokalbių telefonu trukmės minutėmis. Reikia apskaičiuoti, kiek iš viso minučių prakalbėta, ilgiausio pokalbio trukmę, ilgiausių pokalbių skaičių ir kt. Rašant programą bet kurio mėnesio pokalbių telefonu trukmėms apdoroti, negalima žinoti, kiek pokalbių bus tą mėnesį. Todėl negalima iš anksto numatyti, kiek kintamųjų reikia pokalbių trukmėms laikyti. Be to, atskirų pokalbių trukmėms naudojant skirtingus kintamuosius, programa bus gremėzdiška ir neuniversali (kito mėnesio duomenims apdoroti programą reikės taisyti). Masyve visos reikšmės galima laikyti vienu vardu. Reikšmių skaičius masyve gali būti skirtingas.

Masyvo aprašyme nurodomas elementų reikšmių tipas, vardas ir dydis:

```
int A[10];
```

Masyvo dydis

Masyvo vardas

Masyvo elementų reikšmių tipas

Naudojantis masyvu, reikia prisiminti, kad:

- ✓ Visos masyvo elementų reikšmės yra to paties tipo.
- ✓ Norint reikšmę įrašyti į masyvą arba paimti iš jo, reikia nurodyti reikšmės masyve eilės numerį. Tas numeris vadinamas *indeksu*. Jis rašomas už masyvo vardo laužtiniuose skliaustuose.
- ✓ Masyvų elementai vadinami *indeksuotaisiais kintamaisiais*. Jie reiškiniuose ir kitose duomenų struktūrose naudojami taip pat, kaip ir paprasti kintamieji.

Paprastiausiu atveju masyvą galima įsivaizduoti kaip vienodo tipo reikšmių, sunumeruotų iš eilės ir turinčių tą patį vardą, seką. C++ kalboje masyvų elementai pradedami numeruoti nuo nulio.

Pavyzdys

Masyvo A elementų reikšmės atmintyje	15	-3	18	45	25	-4524	-75
Elementų indeksai	0	1	2	3	4	5	6
Kreipinys į masyvo A elementus programavimo kalboje C++	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]

Elemento indeksas gali būti sveikasis skaičius, sveikojo tipo kintamasis arba reiškiny, kurio rezultatas yra sveikasis skaičius. Pavyzdžiui:

```
A[12]; A[i]; A[i + 2]; B[k + 6 * i];
```

Masyvo elementų reikšmių tipas gali būti bazinis (`int`, `double`, `char` ir kt.) arba programuotojo sukurtas.

Masyve galima laikyti ne daugiau reikšmių, negu nurodyta jo apraše. Pavyzdžiui, aprašas `int A[10];` parodo, kad masyve A galima laikyti 10 skaičių: pirmąjį – nuliniame elemente (`A[0]`), antrąjį – pirmame elemente (`A[1]`) ir t. t. Paskutinio elemento indeksas – vienetu mažesnis už masyvo dydį (9). Todėl rašant programas labai svarbu sekti, kad indekso reikšmė visuomet būtų intervale $[0, \text{masyvoDydis} - 1]$.

Programoje galima aprašyti ir naudoti keletą skirtingo tipo ir skirtingo dydžio masyvų. Jų vardai programos bloke turi būti skirtingi.

Masyvo dydis apraše nurodo kompiliatoriui, kiek vietos atmintyje reikia skirti masyvo elementų reikšmėms.

```
int    A[50],    // masyvas, kuriame bus galima laikyti iki 50 sveikųjų skaičių
        C[30];    // masyvas, kuriame bus galima laikyti iki 30 sveikųjų skaičių
double B[60];    // masyvas, kuriame bus galima laikyti iki 60 realiųjų skaičių
```

Aiškinamajame tekste (jei jis rašomas) masyvo dydis paprastai nurodomas už masyvo vardo paprastuosiuose skliaustuose. Pavyzdžiui:

```
A(50), C(30), P(n), D(n * m + 5)
```

Masyvo elementų reikšmės patogiu apdoroti, kai kreipinyje į masyvo elementus indekso vietoje rašomas kintamasis. Pavyzdžiui:

```
int A[50];
for (int i = 0; i < 50; i++) A[i] = i * i;
```

Čia masyve bus surašyti (didėjančiai) skaičių nuo 0 iki 49 kvadratai. Skaičius surašyti mažėjančiai galima pakeitus ciklo antraštę:

```
for (int i = 49; i >= 0; i--) A[i] = i * i;
```

Aprašant masyvą, galima nurodyti jo elementų pradines reikšmes. Pavyzdžiui:

```
int    A[5] = {50, 18, -3, 9, 10};
double B[4] = {1.3, 9.0, 18.34, 9.5};
```

Toks aprašas gali būti funkcijos viduje (lokalieji masyvai) arba už funkcijos `main()` ribų (globalieji masyvai).

Jei programoje masyvo elementams reikšmės nesuteikiamos, tai dažniausiai masyvui skirtoje atminties vietoje įrašomos atsitiktinės reikšmės, vadinamosios šiukšlės. Pavyzdžiui, masyvo `int A[5] = {8, 4, 15};` paskutinės dvi reikšmės `A[3]` ir `A[4]` yra šiukšlės.

Jei masyvo dydis nenurodomas, masyvui atmintyje skiriama tiek vietos, kiek masyvo reikšmių aprašyta:

```
int A[ ] = {6, 89, 21}; // masyvo A dydis bus lygus 3
```

Aprašyti masyvą nenurodant jo dydžio galima tik tada, kai apibrėžiamos masyvo elementų pradines reikšmės. Taigi aprašas `int A[ ];` negalimas.

Masyvo dydis paprastai nurodomas konstanta, kad prireikus jos reikšmę būtų patogiu keisti. Masyvo dydis programoje pasirenkamas toks, kad masyve tilptų didžiausias galimas skaičiavimuose naudojamų reikšmių skaičius. Žinoma, masyvas negali būti begalinis. Jo dydį riboja turimos atmintyje laisvos vietos dydis. Kai reikšmių labai daug ir jų visų vienu metu negalima laikyti masyve, duomenys apdorojami kitais būdais.

Reikšmių priskyrimas masyvo elementams

Masyvo elementams pradinės reikšmės galima suteikti aprašant masyvą arba priskyrimo sakiniiais. Dažniausiai pradiniai duomenys laikomi failuose. Prieš atliekant skaičiavimus su duomenimis, reikia juos iš failo įkelti į masyvą.

Prieš skaitant duomenis iš failo, būtina išsiaiškinti, kaip jie surašyti į failą. Paprastai pirmasis į failą įrašytas skaičius nurodo reikšmių skaičių masyve. Tokiu atveju reikia perskaityti pirmąjį į failą įrašytą skaičių ir patikrinti, ar tiek reikšmių tilps programoje aprašytame masyve. Jei taip, tuomet skaitomos masyvo reikšmės iš failo. Priešingu atveju reikia numatyti galimus tolesnius veiksmus (pvz.: nutraukti programos darbą, perskaityti didžiausią galimą reikšmių skaičių, pasiūlyti naudotojui pakeisti masyvo reikšmių skaičių ir pan.), kurie priklauso nuo papildomų uždavinio sprendimo sąlygų.

Pavyzdys

```
const int Cn = 100;           // masyvo dydis (elementų skaičius)
int A[Cn], n;                 // masyvas ir jo reikšmių skaičius
//-----
// Masyvo D(k) reikšmės skaitomos iš failo, kurio vardas fv
bool Skaityti(char fv[], int D[], int & k)
{
    ifstream fd(fv);
    fd >> k;                   // perskaitomas pirmasis (masyvo reikšmių) skaičius
    if (k > Cn) return false;  // masyve trūksta vietos
    for (int i = 0; i < k; i++)
        fd >> D[i];           // skaitomos masyvo reikšmės
    fd.close();
    return true;               // duomenys sėkmingai perkelti į masyvą
}
```

Duomenų failo pavyzdys

```
12
15 -45 8 4 3 5 8 10
-1 -89 125 14 -85
```

Funkcijos parametrų sąrašė prieš masyvo tipo parametą nereikia rašyti jokių specialių simbolių, nes kreipiantis į funkciją tokiam parametrui visuomet perduodamas argumentas, t. y. nulinio masyvo elemento, adresas. Kreipinys į pavyzdyje pateiktą funkciją gali būti toks:

```
if (Skaityti("Duomenys.txt", A, n)) ...
```

Masyvo reikšmių rašymas

Paprasčiausiai masyvo reikšmės rašyti į failą arba rodyti ekrane.

Pavyzdys

```
// Nuoseklus masyvo reikšmių rašymas
// Masyvo D(k) reikšmės rašomos į failą, kurio vardas fv
void Spausdinti(char fv[], int D[], int k)
{
    ofstream fr(fv);
    for (int i = 0; i < k; i++)
        fr << D[i] << " ";
    fr << endl;
    fr.close();
}
```

Kreipinys į pavyzdyje pateiktą funkciją gali būti toks:

```
Spausdinti("Rezultatai.txt", A, n);
```

Pavyzdyje nurodytos duomenų failo reikšmės į failą bus rašomos nuosekliai viena po kitos:

```
15  -45  8  4  3  5  8  10  -1  -89  125  14  -85
```

Jeigu išvesties srautas yra atvertas, tai jį galima perduoti funkcijai. Tokiu atveju funkcijos antraštėje vietoj failo nurodomas srauto vardas. Parametro tipas turi būti keičiamas į srauto tipą. Srauto kopijos perduoti funkcijai negalima, todėl perduodamas srauto adresas.

Pavyzdys

```
// Nuoseklus masyvo reikšmių rašymas
// Masyvo D(k) reikšmės rašomos į srautą, kurio vardas srautas
void Spausdinti(ofstream & srautas, int D[], int k)
{
    for (int i = 0; i < k; i++)
        srautas << D[i] << " ";
    srautas << endl;
}
```

Čia srautas tik naudojamas. Užverti jį reikia tame programos bloke, kur jis buvo atvertas.

Kreipinys į pavyzdyje pateiktą funkciją galėtų būti toks:

```
Spausdinti(fr, A, n); // fr – išvesties srautas, susietas su failu
Spausdinti(cout, A, n); // cout – išvesties srautas, susietas su ekranu (tekstinis failas)
```

Masyvo perdavimas funkcijai

Masyvo vardas laiko masyvo nulinio elemento adresą, kurio programiškai negalima keisti. Adresas rodo, kur kompiuterio atmintyje prasideda masyvo reikšmių sąrašas. Todėl sakoma, kad masyvo vardas yra rodyklė-konstanta į nulinį masyvo elementą.

Masyvai gali būti funkcijos parametrai. Kai funkcijos parametras yra rodyklė (masyvas), tai visi pakeitimai, atlikti su masyvo elementų reikšmėmis funkcijos kamiene, matomi ir funkcijos išorėje. Norint uždrausti keitimus masyvo viduje, parametną reikia aprašyti su baziniu žodžiu `const`.

Kai funkcijos parametras yra masyvas, tai funkcija iš tikrųjų turi du parametrus: patį masyvą ir masyvo reikšmių kiekį. Funkcijai neužtenka perduoti vieną parametną (patį masyvą). Reikia nurodyti ir masyvo reikšmių kiekį. Jo galima nepateikti tik tam tikrais atskirais atvejais, pavyzdžiui, kai masyvo reikšmių kiekis iš anksto žinomas ir niekada nesikeičia, kai jis įrašytas į nulinį masyvo elementą arba visada yra lygus masyvo dydžiui (elementų skaičiui).

Panagrinėkime dvi funkcijas: vieną masyvo reikšmėms įvesti klaviatūra, kitą – masyvo reikšmėms rodyti ekrane. Funkcijos, skirtos masyvo elementų reikšmėms įvesti, prototipas, pavyzdžiui, yra toks:

```
void Ivesti(int duom[], int & kiek);
```

Čia pirmasis parametras yra masyvas (jo požymis – laužtiniai skliaustai), antrasis – reikšmių kiekis masyve. Antrasis funkcijos parametras kartu yra ir nuoroda, nes tik funkcijos viduje galima sužinoti būsimą masyvo reikšmių kiekį, kurį būtinai reikės perduoti į funkcijos išorę.

Masyvo reikšmių išvedimo funkcijos prototipas gali būti toks:

```
void Isvesti(int duom[], int kiek);
```

Norint kreiptis į funkciją, kurios parametras yra masyvas, kreipinyje tereikia nurodyti masyvo vardą be jokių papildomų ženklų, pavyzdžiui:

```
Ivesti(Am, kiek);  
Isvesti(Am, kiek);
```

```

// Masyvo reikšmių įvedimas klaviatūra
// Masyvas – funkcijos parametras
#include <iostream>
using namespace std;
//-----
bool Ivesti(int Duom[], int & kiek);
void Isvesti(int Duom[], int kiek);
//-----
const int Cilgis = 10;
int main()
{
    int Am[Cilgis], Akiek;
    bool sekme = Ivesti(Am, Akiek);
    if (sekme)
        Isvesti(Am, Akiek);
    else {
        cout << "Per daug elementų paprašėte įvesti" << Akiek << endl;
        cout << "Galima tik " << Cilgis;
    }
    return 0;
}
//-----
// Įvedamos klaviatūra masyvo Duom(kiek) reikšmės
bool Ivesti(int Duom[], int & kiek)
{
    cout << "Įveskite elementų skaičių ";
    cin >> kiek;
    if (kiek > Cilgis)
        return false;
    cout << "Įveskite " << kiek << " masyvo elementus (-ų)"
        << endl;
    for (int i = 0; i < kiek; i++)
        cin >> Duom[i];
    return true;
}
//-----
// Rodomos ekrane masyvo Duom(kiek) reikšmės
void Isvesti(int Duom[], int kiek)
{
    cout << " Masyvo elementai " << endl;
    for (int i = 0; i < kiek; i++)
        cout << Duom[i] << " ";
    cout << endl;
}

```

Programos darbo rezultatas

```
Įveskite elementų skaičių 5  
Įveskite 5 masyvo elementus (-ų)  
10 11 12 54 8
```

Realizuodami funkciją `Ivesti()`, šiek tiek pakeitėme jos antraštę. Dabar funkcija grąžina pranešimą `true`, jei duomenys įvesti į masyvą, arba `false`, jei neįvesti. Pagrindinėje funkcijoje tikrinama, ar pavyko įvesti. Tai leidžia lengviau aptikti programos klaidas. Programoje aprašytas masyvas yra fiksuoto dydžio. Programos darbo metu klaviatūra nurodoma, kiek reikšmių reikia įvesti į masyvą. Jeigu prašoma įvesti daugiau, nei gali tilpti masyve, tai funkcija `Ivesti()` grąžina reikšmę `false`. Tai rodo, kad į masyvą reikšmės nėra įvestos. Jeigu įvesties sraute yra daugiau reikšmių, nei gali tilpti masyve, tuomet veiksams su visomis reikšmėmis atlikti būtina taikyti kitus duomenų apdorojimo algoritmus.

Pavyzdinė užduotis

Krituliai. Lietuvos hidrometeorologijos tarnyba kaupia įvairiose Lietuvos vietovėse vasarą iškritusių kritulių stebėjimų duomenis.

Parenkite programą, kuri pateiktų informaciją apie lietingas dienas:

- ✓ kiek milimetrų kritulių iškrito iš viso;
- ✓ kiek dienų nelijo;
- ✓ kiek milimetrų kritulių vidutiniškai iškrito tomis dienomis, kai lijo.

Pirmoje failo eilutėje yra nurodytas stebėtų dienų skaičius. Antroje eilutėje surašyti kiekvieną stebėtą dieną iškritusių kritulių kiekiai (milimetrais). Jeigu kuria nors dieną nelijo, tai kritulių kiekis tą dieną lygus 0.

Pradinių duomenų ir rezultatų failų pavyzdys

[illegible]

Algoritmas

Užduotis sprendžiama taip:

1. Iš failo į masyvą skaitomi pradiniai duomenys.
2. Masyvo reikšmės rašomos į failą lentelė: diena, kritulių kiekis.
3. Atliekami skaičiavimai:
 - ✓ kiek milimetrų kritulių iškrito iš viso;
 - ✓ kiek dienų nelijo;
 - ✓ kiek milimetrų kritulių vidutiniškai iškrito tomis dienomis, kai lijo.
4. Apskaičiuoti rezultatai rašomi į failą.

Programos struktūra

Funkcijos pavadinimas	Funkcijos paskirtis
Skaityti()	Pradinių duomenų skaitymas iš failo į masyvą
Spausdinti()	Masyvo reikšmių rašymas į failą lentelė
KrituliuKiekis()	Kritulių kiekio (sumos) skaičiavimas
DienuSkaicius()	Nelietingų dienų skaičiaus radimas
KrituliuVidurkis()	Lietingomis dienomis iškritusių kritulių vidurkio skaičiavimas



Pradinių duomenų failo kūrimas, konstantų ir kintamųjų aprašymas

- Sukurkite tekstinį failą *Duomenys6.txt* ir į jį įrašykite pateiktus pavyzdyje pradinis duomenis.
- Prieš pagrindinę funkciją `main()` aprašykite konstantas pradinių duomenų ir rezultatų failų vardams bei masyvo dydžiui atmintyje laikyti.
- Pagrindinėje funkcijoje aprašykite sveikųjų skaičių masyvą `Kr(n)` duomenims – kritulių kiekams – atmintyje laikyti.

```
const char CDfv[] = "Duomenys6.txt";    // pradinių duomenų failo vardas
const char CRfv[] = "Rezultatai6.txt";  // rezultatų failo vardas
const int  CMax   = 100;                // masyvo dydis
//-----
// Funkcijų prototipai
//-----
int main()
{
    int Kr[CMax]; int n;                // kritulių kiekių masyvas ir jame laikomų reikšmių kiekis
    return 0;
}
```

- Įrašykite ir įvykdysite programą. Ekrane turėtumėte matyti tik informacinį pranešimą.



Pradinių duomenų skaitymas iš failo į masyvą $Kr(n)$

- Sukurkite funkciją `Skaityti()`, kuri iš duomenų failo skaitytų pradinis duomenis.
- Parašykite funkcijos `Skaityti()` prototipą:

```
void Skaityti(int A[], int & n);
```

čia parametras A skirtas sveikųjų skaičių masyvui, o n – jame laikomų reikšmių skaičiui įsiminti.

- Parašykite funkcijos `Skaityti()` tekstą:

```
// Skaito duomenis iš failo CDfv į masyvą A(n)
void Skaityti(int A[], int & n)
{
    ifstream fd(CDfv);           // atidaromas įvesties srautas
    fd >> n;                      // perskaitomas masyvo reikšmių kiekis
    for (int i = 0; i < n; i++)
        fd >> A[i];              // perskaitoma i-oji reikšmė
    fd.close();                  // uždaromas įvesties srautas
}
```

- Funkcijoje `main()` parašykite kreipinį į funkciją `Skaityti()` ir perskaitytos kintamojo n reikšmės rodymo ekrane sakinius:

```
Skaityti(Kr, n);
cout << n << endl;
```

- Įrašykite ir įvykdykite programą. Ekrane matysite stebėtų dienų skaičių:

```
10
```

Į masyvą $Kr(n)$ iš failo buvo perskaityta 10 reikšmių.

- Iš programos pašalinkite eilutę, kad kintamojo n reikšmė nebūtų rodoma ekrane.

3

Masyvo Kr (n) reikšmių (kritulių kiekių) rašymas į failą lentele

- Norint peržiūrėti, kokie duomenys buvo perskaityti iš failo į masyvą, būtina juos įrašyti. Sukurkite funkciją Spausdinti(), kuri įrašytų lentele masyvo Kr (n) reikšmes į rezultatų failą.
- Parašykite funkcijos Spausdinti() prototipą:

```
void Spausdinti(int A[], int n);
```

- Parašykite funkcijos Spausdinti() tekstą:

```
// Rašo lentele masyvo A(n) reikšmes į failą CRfv
void Spausdinti(int A[], int n)
{
    ofstream fr(CRfv);           // atidaromas išvesties srautas
    fr << "    Krituliai (lietus)    " << endl;
    fr << "-----" << endl;
    fr << "Diena  Kritulių kiekis (mm)" << endl;
    fr << "-----" << endl;
    for (int i = 0; i < n; i++)
        fr << setw(4) << i+1 << "    " << setw(3) << A[i] << endl;
    fr << "-----" << endl;
    fr.close();                  // uždaromas išvesties srautas
}
```

- Funkcijoje main() parašykite kreipinį į funkciją Spausdinti():

```
Spausdinti(Kr, n);
```

- Įrašykite ir įvykdysite programą.
- Atverkite rezultatų failą Rezultatai6.txt. Jame pamatysite kritulių kiekių lentelę:

Krituliai (lietus)	

Diena	Kritulių kiekis (mm)

1	10
2	20
3	0
4	0
5	0
6	45
7	25
8	30
9	50
10	25



Viso kritulių kiekio (sumos) skaičiavimas

- Parašykite funkcijos `KrituliuKiekis()` prototipą:

```
int KrituliuKiekis(int A[], int n);
```

- Parašykite funkcijos `KrituliuKiekis()` tekstą:

```
// Apskaičiuoja ir grąžina masyvo A(n) reikšmių sumą
int KrituliuKiekis(int A[], int n)
{
    int suma = 0;
    for (int i = 0; i < n; i++)
        suma = suma + A[i];
    return suma;
}
```

- Papildykite funkciją `main()` kintamaisiais `kk` ir `fr`:

```
int kk;                // visas kritulių kiekis (suma)
ofstream fr;           // išvesties srautas
```

- Funkcijoje `main()` atverkite papildyti išvesties srautą `fr`:

```
fr.open(CRf, ios::app); // atidaro išvesties srautą papildyti
```

- Parašykite kreipinį į funkciją `KrituliuKiekis()`:

```
kk = KrituliuKiekis(Kr, n);
```

- Įrašykite gautą rezultatą į rezultatų failą, o po to užverkite išvesties srautą:

```
fr << "Iš viso iškrito kritulių (mm): " << kk << endl;
fr.close(); // uždaro išvesties srautą
```

- Įrašykite ir įvykdykite programą. Atverkite rezultatų failą *Rezultatai6.txt*. Jame turėtų būti papildoma eilutė:

```
Iš viso iškrito kritulių (mm): 205
```



Nelietingų dienų skaičiavimas

- Parašykite funkcijos `DienuSkaicius()` prototipą:

```
int DienuSkaicius(int A[], int n);
```

- Parašykite funkcijos `DienuSkaicius()` tekstą:

```
// Apskaičiuoja, kiek masyve A(n) yra nulinių reikšmių, ir jas grąžina
int DienuSkaicius(int A[], int n)
{
    int kiek = 0;
    for (int i = 0; i < n; i++)
        if (A[i] == 0)
            kiek = kiek + 1;
    return kiek;
}
```

- Papildykite funkciją `main()` kintamuoju `kd`:

```
int kd; // kiek dienų nelijo
```

- Funkcijoje `main()` parašykite kreipinį į funkciją `DienuSkaicius()`:

```
kd = DienuSkaicius(Kr, n);
```

- Įrašykite gautą rezultatą į failą:

```
fr << "Nelijo (dienas): " << kd << endl;
```

- Įrašykite ir įvykdykite programą. Atverkite rezultatų failą *Rezultatai6.txt*. Jame turėtų būti papildoma eilutė:

```
Nelijo (dienas): 3
```



Lietingomis dienomis iškritusių kritulių vidurkio skaičiavimas

- Parašykite funkcijos `KrituliuVidurkis()` prototipą:

```
int KrituliuVidurkis(int A[], int n);
```

- Parašykite funkcijos `KrituliuVidurkis()` tekstą:

```
// Apskaičiuoja masyvo A(n) teigiamųjų reikšmių vidurkį ir jį grąžina
// Jeigu masyve teigiamųjų reikšmių nebuvo, grąžina 0 (nulį)
int KrituliuVidurkis(int A[], int n)
{
    int suma = 0;
    int kiek = 0;
    for (int i = 0; i < n; i++)
        if (A[i] > 0) {
            suma = suma + A[i];
            kiek = kiek + 1;
        }
    if (kiek > 0)
        return suma / kiek;
    return 0;
}
```

- Papildykite funkciją `main()` kintamuoju `vid`:

```
int vid;           // kritulių vidurkis lietingomis dienomis
```

- Funkcijoje `main()` parašykite kreipinį į funkciją `KrituliuVidurkis()`:

```
vid = KrituliuVidurkis(Kr, n);
```

- Įrašykite gautą rezultatą į rezultatų failą:

```
fr << "Vidutiniškai kiekvieną lietingą dieną iškrito kritulių (mm): "  
<< vid << endl;
```

- Įrašykite ir įvykdykite programą.

- Atverkite rezultatų failą *Rezultatai6.txt*. Jame pamatysite visus sukurtos programos rezultatus:

```
      Krituliai (lietus)  
-----  
Diena  Kritulių kiekis (mm)  
-----  
    1         10  
    2         20  
    3          0  
    4          0  
    5          0  
    6         45  
    7         25  
    8         30  
    9         50  
   10         25  
-----  
Iš viso iškrito kritulių (mm): 205  
Nelijo (dienas): 3  
Vidutiniškai kiekvieną lietingą dieną iškrito kritulių (mm): 29
```

Pagrindinė funkcija

Nuosekliai atlikus visus šio darbo žingsnius, pagrindinė funkcija turėtų būti tokia:

```
const char CDfv[] = "Duomenys6.txt";    // pradinių duomenų failo vardas
const char CRfv[] = "Rezultatai6.txt";   // rezultatų failo vardas
const int  CMax  = 100;                  // masyvo dydis
//-----
void Skaityti(int A[], int & n);
void Spausdinti(int A[], int n);
int KrituliuKiekis(int A[], int n);
int DienuSkaicius(int A[], int n);
int KrituliuVidurkis(int A[], int n);
//-----
int main ()
{
    int Kr[CMax]; int n; // kritulių kiekių masyvas
    int kk;              // visas kritulių kiekis (suma)
    int kd;              // kiek dienų nelijo
    int vid;             // kritulių vidurkis lietingomis dienomis
    ofstream fr;         // išvesties srautas
    Skaityti(Kr, n);
    Spausdinti(Kr, n);
    fr.open(CRfv, ios::app); // atidaro išvesties srautą papildyti
    kk = KrituliuKiekis(Kr, n);
    fr << "Iš viso iškrito kritulių (mm): " << kk << endl;
    kd = DienuSkaicius(Kr, n);
    fr << "Nelijo (dienas): " << kd << endl;
    vid = KrituliuVidurkis(Kr, n);
    fr << "Vidutiniškai kiekvieną lietingą dieną iškrito kritulių (mm): "
        << vid << endl;
    fr.close();          // uždaro išvesties srautą
    return 0;
}
```



Programos patikrinimas

- Paruoškite skirtingų duomenų rinkinius ir patikrinkite, ar programa visais atvejais pateikia teisingus rezultatus. Siūlome patikrinti programą tais atvejais, kai:
 - ✔ kiekvieną stebėtą dieną lijo;
 - ✔ nė vieną stebėtą dieną nelijo;
 - ✔ iškritusių kritulių reikšmės pradinio duomenų faile surašytos keliose eilutėse.



Programos papildymas

- Papildykite programą, kad ji apskaičiuotų:
 - ✔ kiek dienų lijo;
 - ✔ kiek vidutiniškai kritulių iškrito per visas stebėtas dienas.

Kiekvienu atveju pasiruoškite kontrolinių duomenų rinkinius. Iš anksto apskaičiuokite, kokie turėtų būti rezultatai.

Užduotys



Užduotys

1. Gyventojai

Pirmoje pradinių duomenų failo eilutėje įrašytas natūralusis skaičius n ($n < 500$). Jis nurodo, kiek gatvėje yra namų. Toliau yra n eilučių, kurių kiekvienoje yra du tarpais atskirti sveikieji skaičiai: namo numeris ir gyventojų tame name skaičius. Namai, kurių numeriai lyginiai, stovi dešiniojoje, o kurių nelyginiai – kairiojoje gatvės pusėje. Parenkite programą, kuri apskaičiuotų ir į rezultatų failą įrašytų:

- ✓ kiek gyventojų iš viso gyvena gatvėje;
- ✓ kiek gyventojų gyvena kairiojoje gatvės pusėje ir kiek – dešiniojoje;
- ✓ kiek vidutiniškai gyventojų gyvena kiekviename name, esančiame kairiojoje gatvės pusėje, ir kiek – dešiniojoje.

Pradiniai duomenys	Rezultatai	Paaiškinimai
10	34	Iš viso gyventojų gatvėje
1 5	20	Iš viso gyventojų kairiojoje gatvės pusėje
2 4	14	Iš viso gyventojų dešiniojoje gatvės pusėje
3 4	4.00	Vidutinis gyventojų skaičius name, esančiame kairiojoje gatvės pusėje
4 5	2.80	Vidutinis gyventojų skaičius name, esančiame dešiniojoje gatvės pusėje
5 6		
6 5		
7 0		
8 0		
9 5		
10 0		

2. Ūgiai

Pirmoje pradinių duomenų failo eilutėje nurodytas klasės mokinių skaičius n ($n < 30$). Tolesnėse n eilučių įrašytas kiekvieno klasės mokinio ūgis centimetrais. Merginų ūgiai yra teigiamieji, vaikinų – neigiamieji sveikieji skaičiai. Parenkite programą, kuri apskaičiuotų ir į rezultatų failą įrašytų:

- ✓ klasės mokinių vidutinį ūgį 0,1 cm tikslumu;
- ✓ klasės merginų vidutinį ūgį 0,01 cm tikslumu;
- ✓ klasės vaikinų vidutinį ūgį 0,01 cm tikslumu;
- ✓ pranešimą, ar galima sudaryti iš klasės merginų ir vaikinų krepšinio komandas. Reikalavimai krepšinio komandai: ūgis ne mažesnis negu 175 cm, komandoje turi būti 7 žaidėjai (5 pagrindiniai ir 2 atsarginiai).

Pradiniai duomenys	Rezultatai
13	175.8
-178	172.57
175	179.67
-186	Merginų komandos sudaryti negalima
172	Vaikinų komandos sudaryti negalima
173	
175	
-185	
-180	
-169	
165	
176	
172	
-180	

3. Skaitytojai

Pirmoje pradinių duomenų failo eilutėje nurodytas klasės mokinių skaičius n ($n < 30$). Tolesnėse n eilučių įrašyta po vieną sveikąjį skaičių – kiek kiekvieno mokinio per mokslo metus (10 mėnesių) perskaityta knygų. Parenkite programą, kuri į rezultatų failą įrašytų:

- ✓ kiek iš viso knygų per mokslo metus perskaitė mokiniai;
- ✓ kiek vidutiniškai knygų per mokslo metus perskaitė vienas mokinys (rezultatas apvalinamas iki sveikąjo skaičiaus);
- ✓ kiek vidutiniškai knygų per mėnesį perskaitė vienas mokinys (rezultatas pateikiamas vieno ženklo po kablelio tikslumu).

Pradiniai duomenys	Rezultatai
5	65
12	13
13	1.3
12	
15	
13	

4. Garso signalai

Garso signalas gali būti koduojamas sveikųjų skaičių seka. Šie skaičiai rodo signalo stiprumą periodiniais laiko intervalais. Signalą iškraipantis triukšmas šiek tiek pakeičia tų skaičių reikšmes.

Signalą „išlyginimo“ metu triukšmas pašalinamas tokiu būdu: kiekvienas skaičius keičiamas jo ir dviejų jam gretimų skaičių vidurkiu (vidutinės reikšmės sveikąja dalimi). Pirmas ir paskutinis skaičiai atitinkamai keičiami dviejų pirmųjų arba dviejų paskutinių skaičių vidurkiu.

Faile yra n ($0 < n < 50$) garso signalus atitinkančių skaičių sekų. Kiekviena seka sudaryta iš k ($1 < k < 50$) sveikųjų skaičių ir užrašyta vienoje failo eilutėje, skaičiai atskirti tarpais. Pirmoje failo eilutėje yra n ir k reikšmės, atskirtos tarpais.

Į rezultatų failą surašykite „išlygintus“ garso signalus atitinkančias skaičių sekas atskiromis eilutėmis.

Pradiniai duomenys	Rezultatai
3 5	5 4 5 5 6
4 7 3 5 8	8 8 8 7 7
8 9 7 8 6	4 5 5 6 6
5 4 6 7 6	

Sudėtingesnės užduotys

5. Akmenukai

Eilėje stovi n vaikų. Jie sunumeruoti (iš kairės į dešinę) iš eilės nuo 1 iki n . Kiekvienas vaikas arba turi saują akmenukų, arba sauja yra tuščia. Vaikai paeiliui dalija akmenukus stovintiems dešinėje vaikams.

Pirmasis vaikas duoda vieną akmenuką antrajam, po to – vieną akmenuką trečiajam ir t. t. Taip dalydamas akmenukus, jis eina į dešinią eilės galą, po to atgal link savo vietos kiekvienam vaikui vėl duodamas po akmenuką. Dešiniajame eilės gale stovintis (n -asis) vaikas vis gauna po akmenuką, kai dalijantis akmenukus vaikas eina į priekį ir atgal.

Vaikas nustoja dalyti akmenukus, kai jis prieina savo vietą arba pasibaigia akmenukai. Jeigu vaikui pasibaigia akmenukai dar nepriejus savo vietos, jis grįžta tuščiomis rankomis. Jeigu vaikas prieina savo vietą turėdamas rankose akmenukų, jis atsistoja ir pasilieka likusius akmenukus.

Toliau analogiškai akmenukus dalija antrasis vaikas, trečiasis ir t. t. Paskutinis (n -asis) vaikas, stovintis dešiniajame eilės gale, akmenukų nedalija.

Akmenukai dalijami tik dešinėje stovintiems vaikams, todėl antrasis vaikas duoda akmenukų tik trečiajam, ketvirtajam ir t. t., trečiasis – ketvirtajam, penktajam ir t. t.

Parenkite programą, kuri apskaičiuotų, kiek akmenukų turės kiekvienas vaikas, kai baigsis dalijimas.

Pirmoje pradinių duomenų failo eilutėje nurodytas vaikų skaičius ($3 \leq n \leq 10$). Antroje eilutėje įrašyta n tarpais atskirtų sveikųjų neneigiamųjų skaičių – kiekvieno vaiko turimas akmenukų skaičius.

Rezultatų failo vienoje eilutėje turi būti pateikiami kiekvieno vaiko turimi akmenukų skaičiai, atskirti tarpais.

Pradiniai duomenys	Rezultatai
5	0 1 0 15 16
3 6 1 12 10	

Atostogos

Per vasaros atostogas n šeimų ruošiasi keliauti po Europą skirtingų maršrutų greitaisiais traukiniais. Keliauti planuoja kiekvienos šeimos s suaugusiųjų ir v vaikų. Bilietas suaugusiajam kainuoja sk litų, vaikui – vk litų. Parenkite programą, kuri apskaičiuotų, kiek kiekvienos šeimos narių zs ruošiasi keliauti ir kokios numatomos kelionės išlaidos kk .

Pirmoje pradinių duomenų failo *Duomenys.txt* eilutėje įrašytas šeimų skaičius n . Tolesnėse n eilučių įrašyta po du sveikuosius skaičius: vykstančių į kelionę šeimos suaugusiųjų ir vaikų skaičius, ir po du realiuosius skaičius – bilieto suaugusiajam ir vaikui kaina.

Rezultatų failo *Rezultatai.txt* atskirose eilutėse pateikite informaciją apie kiekvieną šeimą: kiek iš viso narių zs ruošiasi keliauti ir kelionės išlaidas kk dviejų skaitmenų po kablelio tikslumu.

Nurodymai

- naudokite sveikųjų ir realiųjų skaičių masyvus;
- sukurkite pradinių duomenų skaitymo funkciją `void`;
- sukurkite rezultatų rašymo į failą funkciją `void`.

Pradiniai duomenys	Rezultatai
4	4 454.30
2 2 147.00 80.15	3 354.00
1 2 163.00 95.50	3 374.15
2 1 147.00 80.15	2 258.50
1 1 163.00 95.50	

Programos patikrinimas

Pradiniai duomenys				Rezultatai	
Suaugusiųjų skaičius	Vaikų skaičius	Bilieto kaina suaugusiajam	Bilieto kaina vaikui	Keliaujančiųjų skaičius	Kelionės kaina kk
s	v	sk	vk	$s + v$	
1	0	10	5	1	10.00
1	10	10	5	11	60.00
10	10	10	0	20	100.00
0	5	10	10	5	50.00

Automobiliai

Gintaras nori pirkti „Toyota“ markės automobilį. Norėdamas atsirinkti tinkamus interneto parduotuvės pasiūlymus, jis pateikė užklausą ir nurodė automobilio pagaminimo metus pm , variklio darbinį tūrį v cm^3 ir pardavimo kainą ak . Parenkite programą, kuri apibendrintų pagal užklausą pateiktą informaciją ir sukurtų tokį automobilių sąrašą, kuriame būtų nurodyta: eilės numeris, automobilio pagaminimo metai, variklio darbinis tūris, pardavimo kaina, naujo automobilio kaina (ji apskaičiuojama įvertinant automobilio nusidėvėjimą).

Pradinių duomenų failo *Duomenys.txt*:

- pirmoje eilutėje įrašytas „Toyota“ markės automobilių modelių skaičius n ;
- tolesnėse n eilučių nurodyta po du sveikuosius skaičius – modelio pagaminimo metai bei variklio darbinis tūris, ir vienas realusis skaičius – automobilio pardavimo kaina;
- paskutinėje eilutėje įrašyti automobilio pardavimo metai m ir per kiek metų p nusidėvi automobilis.

Rezultatų failo *Rezultatai.txt* atskirose eilutėse pateikite šią informaciją apie automobilius:

- eilės numerį;
- pagaminimo metus;
- variklio darbinį tūrį;
- kainą ak metais m ;
- naujo automobilio kainą, kuri apskaičiuojama įvertinant nusidėvėjimą: $ak / ((m - pm) / p)$.

Nurodymai

- naudokite sveikųjų ir realiųjų skaičių masyvus;
- sukurkite pradinių duomenų skaitymo funkciją `void`;
- sukurkite pradinės kainos skaičiavimo funkciją;
- sukurkite rezultatų rašymo į failą funkciją `void`.

Pradiniai duomenys	Rezultatai
4	1 2003 2000 19000.00 31666.67
2003 2000 19000.00	2 2003 2000 19200.00 32000.00
2003 2000 19200.00	3 2005 1400 15020.00 32185.71
2005 1400 15020.00	4 2003 2000 13462.00 22436.67
2003 2000 13462.00	
2012 15	

Daržovės

Augustė žinyne surado informacijos apie daržovių energinę vertę ir nusprendė parengti programą, kuri apskaičiuotų, kiek angliavandenių, baltymų, riebalų ir vandens yra kiekvienos daržovės 100 gramų.

Pradinių duomenų failo *Duomenys.txt*:

- pirmoje eilutėje įrašyti trys sveikieji skaičiai ak , bk , rk – 1 gramų angliavandenių, baltymų ir riebalų kilokalorijų (kcal) kiekiai;
- antroje failo eilutėje įrašytas skirtingų daržovių skaičius n ;
- tolesnėse n failo eilučių įrašyta po tris realiuosius skaičius – 100 gramų daržovių sudedamųjų dalių (angliavandenių a , baltymų b ir riebalų r) energinė vertė kcal.

Į rezultatų failą *Rezultatai.txt* surašykite kiekvienos daržovės 100 gramų angliavandenių, baltymų, riebalų ir vandens kiekius gramais. Sakykime, kad daržovės sudarytos tik iš nurodytų sudedamųjų dalių.

Nurodymai

- naudokite realiųjų skaičių masyvus;
- sukurkite pradinių duomenų skaitymo funkciją `void`;
- sukurkite daržovės kiekvienos sudedamosios dalies kiekio skaičiavimo funkciją;
- sukurkite rezultatų rašymo į failą funkciją `void`.

Pradiniai duomenys	Rezultatai
4 4 9	2.8 0.7 0.0 96.5
4	5.1 1.7 26.4 66.8
11.2 2.8 0.0	6.6 0.8 0.2 92.4
20.4 6.8 237.6	5.0 2.9 0.3 91.8
26.4 3.2 1.8	
20.0 11.6 2.7	