Programavimo C++ pradmenys

III klasė

Nežinomo kartojimų skaičiaus ciklas WHILE

Mokymosi tikslas ir uždaviniai:

Suprasti ir gebėti taikyti WHILE ciklą, kai iš anksto nėra žinomas kartojimų skaičius, sprendžiant programavimo užduotis.

- *susipažinti su WHILE ciklo veikimo principu ir jo struktūra;*
- *gebėti atskirti WHILE ciklą nuo FOR ciklo, pabrėžiant jų taikymo skirtumus;*
- *gebėti analizuoti ir taisyti klaidas, susijusias su WHILE ciklo naudojimu (begaliniai ciklai, neteisinga sąlyga ir pan.).*
- *įtvirtinti skaitymo rašymo konsolėje įgūdžius;*
- *pakartoti ir suvokti max/min radimo, sumos ir kiekio skaičiavimo algoritmus;*
- *įgytas žinias ir įgūdžius pritaikyti sprendžiant uždavinius.*

Kartais rašant programas norisi tam tikrus veiksmus kartoti daug kartų. Tuomet naudojami ciklai. C++ programavimo kalba palaiko trijų rūšių ciklus:

- for
- while
- do...while

Žinomo kartojimų skaičiaus ciklas FOR rašomas tuomet, kai žinome, kiek kartų reikia kartoti vieną ar kelis veiksmus (ciklas su skaitliuku). Labai dažnai ***veiksmai kartojami tol, kol tenkinama nurodyta sąlyga***. Pavyzdžiui vykdome skaičiavimus ir norime juos vykdyti iki kol pasieksime tinkamą rezultatą. Tokiais atvejais naudojamas ciklo sakinys **`while`** (ciklas su sąlyga). Jeigu reikia kartoti kelis sakinius, jie rašomi tarp riestinių skliaustų {}. Šio ciklo sintaksė:

kai kartojamas vienas veiksmas	**kai kartojami keli veiksmai**

`while (Sąlyga)`
`  Kartojamas sakinys, kai sąlyga tenkinama;`
	`while (Sąlyga)`
	`  {`
	`  Kartojami sakiniai, kai sąlyga tenkinama;`
	`  }`

Sąlyga – loginis reiškinys. Jei to loginio reiškinio reikšmė teisinga, tai vykdomi *Kartojami sakiniai*, kuriais dažniausiai reikia atlikti skaičiavimus ar išvesti kokius nors pranešimus bei keisti sąlygoje esančių kintamųjų reikšmes, kad sąlygos loginis reiškinys kada nors įgautų klaidingą reikšmę ir ciklas būtų užbaigtas.

Reikėtų nepamiršti prieš ciklą nusakyti pradines (galutines) reikšmes, kurios dažniausiai keičiamos sąlygoje atliekant ciklą.

Jei ciklo antraštėje užrašyta sąlyga netenkinama, tuomet veiksmai cikle neatliekami, atliekami tolesni po ciklo sakiniu einantys veiksmai.

Jei ciklo antraštėje uţrašyta sąlyga visada tenkinama, tuomet ciklas atliekamas be galo daug kartų ir vadinamas *amžinuoju ciklu.*

Panagrinėkime programos fragmentus::

1. duotasis x (pvz. 10) didinamas dvejetu kol pasiekia nurodytą reikšmę (pvz. 15). Pateikiama gauta x reikšmė.

```
...
x = 10;
while (x <= 15)
   x = x + 2;
…
Kelintą kartą kartojamas sakinys        x <= 15                    x = x + 2;
1                                       10 <= 15                   10 + 2 = 12
2                                       12 <= 15                   12 + 2 = 14
3                                       14 <= 15                   14 + 2 = 16
4                                       16 <= 15 - sąlyga netenkinama, sakinys nekartojamas
Sakinys pakartotas 3 kartus. x = 16.
```

vietoje *x=x+2* galima rašyti *x += 2;*

2. duotasis x (pvz. 10) didinamas dvejetu kol pasiekia nurodytą reikšmę (pvz. 15), o duotasis y (pvz. 5) mažinamas vienetu. Pateikiamos gautos x ir y reikšmės.

```
...
x = 10; y = 5;
while (x <= 15){
   x = x + 2;
   y = y - 1;
}

Kelintą kartą kartojami sakiniai
              x <= 15                  x = x + 2;                 y = y - 1;
1             10 <= 15                 10 + 2 = 12                5 - 1 = 4
2             12 <= 15                 12 + 2 = 14                4 - 1 = 3
3             14 <= 15                 14 + 2 = 16                3 - 1 = 2
4             16 <= 15 - sąlyga netenkinama, sakiniai nekartojami
Sakinys pakartotas 3 kartus. x = 16, y = 2.
```

3. Spausdinami teigiami skaičiai mažėjančiai nuo įvestojo n




```
…
cin >> n;
while (n>0)
   {
      cout << n << " ";
      n = n - 1; //galima rašyti n--
   }
…
```

4. Duoti skaičiai a ir b (pvz. 10 ir 5). Kol a išlieka didesnis už b, skaičius a mažinamas vienetu, o skaičius b didinamas vienetu ir spausdinamos gautos a ir b reikšmės.

```
…
a = 10; b = 5
While (a > b)
{
   a = a - 1; //galima rašyti a--
   b = b + 1; //galima rašyti b++
   cout << a << b << endl;
}
…
```

kol a>b		
a=a-1	*b=b+1*	
a=10	b=5	*10>5*
10-1=9	5+1=6	*9>6*
9-1=8	6+1=7	*8>7*
8-1=7	7+1=8	*7>8*

5. Duoti skaičiai a ir b (pvz. 10 ir 5). Kol a išlieka didesnis už b, skaičius a didinamas vienetu, o skaičius b mažinamas vienetu ir spausdinamos gautos a ir b reikšmės.

```
…
a = 10; b = 5
While (a > b)
{
   a = a + 1;
   b = b - 1;
   cout << a << b << endl;
}
…
```

a	b
11	5
12	4
…	…

amžinasis ciklas

Jei ciklas amžinas, programos vykdymą galima nutraukti klavišų kombinacija *CTRL+C*

Kaupiamasis4.2.WHILE *Skaičių sekos.*

Pavyzdinė programa1 ***Parašyti programą, kuri apskaičiuotų įvedamos skaičių sekos sumą. Tarkime, kad tarp sekos skaičių nulio nėra. Sekos pabaigoje yra nulis, kuris yra sekos pabaigos požymis.***

```
7   using namespace std;
8
9   int main()
10  {
11   setlocale(LC_ALL, "Lithuanian");
12
13    int n, sk, sum;
14    sum = 0;    //priskiria pradinė reikšmė
15
16    //cout<<"Įvedama sveikųjų skaičių seka. Pabaigos požymis - nulis"<<endl;
17    //cout<<"Įveskite pirmąjį sekos skaičių"<<endl;
18    cin >> sk;    // pirmasis sekos skaičius
19
20    while (sk != 0)     //ciklas tikrina salyga
21     {
22       sum = sum + sk;    //kaupia skaičių suma
23       //cout<<"Įveskite sekantį sekos skaičių"<<endl;
24       cin >> sk;    //perskaito kitą sekos skaičių
25     }
26
27    cout << "Skaičių sekos suma = " << sum << endl;
28   return 0;
29  }
```

```
5
2
10
4
0
Skaičių sekos suma = 21
```

P.S. pavyzdyje dalis spausdinimo sakinių “užkomentuoti”, nes nėra būtini, bet gali būti vartojami (nuėmus komentaro ženklą).

Pavyzdinė programa2 ***Parašykite programą, kuri suskaičiuotų KIEK natūraliųjų skaičių intervale [a; b] yra lyginių skaičių, besidalijančių iš 3. Tokius skaičius reikia išvesti į ekraną. Naudoti while***

```
int a, b, kiek;

cout << "Iveskite intervalo pradzia" << endl;
cin >> a;

cout << "Iveskite intervalo pabaiga (reiksme didesne nei intervalo pradzios)" << endl;
cin >> b;

kiek = 0;

while (a <= b)
{
    if (a % 2 == 0 && a % 3 == 0)
    {
        cout << a << " ";
        kiek++;
    }

    a++;   // kadangi kintamojo a reikšmė jau panaudota, ją reikia keisti, kad ciklo sąlyga keistųsi
}

cout << endl;
cout << "Lyginiu skaiciu, daliu is 3, nurodytame intervale yra: " << kiek << endl;
```

kurta DI

Uždavinių sprendimas įgūdžiams formuoti_1

1. ***NeigSuma***

Klaviatūra įvedama daug skaičių. Parašyti programą, kuri apskaičiuotų sekos neigiamųjų skaičių sumą. Tarkime, kad tarp sekos skaičių nulio nėra. Sekos pabaigoje yra nulis, kuris yra sekos pabaigos požymis

2. ***KiekSkaiciu***

Klaviatūra įvedama daug skaičių. Parašyti programą, kuri apskaičiuotų, kiek skaičių sudaro seką. Sutarkime, kad sekos pabaigos požymis yra nulis.

3. ***Vidurkis***

Klaviatūra įvedama n skaičių. Parašyti programą, kuri rastų įvestų skaičių sumą ir vidurkį šimtųjų tikslumu. Naudoti While ciklą.

4. ***Lyginumas***

Klaviatūra įvedama daug skaičių. Parašyti programą, kuri apskaičiuotų sekos visų lyginių ir visų nelyginių skaičių sumas. Tarkime, kad tarp šių sekos skaičių nulio nėra. Sekos pabaigoje yra nulis, kuris yra sekos pabaigos požymis

5. ***Intervalas***

Parašyti programą, kuri apskaičiuotų skaičių, kurie priklauso intervalui (n; m] sandaugą ir suskaičiuotų kiek tarp jų yra lyginių. Naudoti While ciklą. Sutarkime, kad sekos pabaigos požymis yra nulis. Tarkime, kad tarp šių sekos skaičių nulio nėra.

6. ***Ugiai***

Klaviatūra įvedama daug skaičių, kurie nurodo klasės mokinių ūgį (cm). Pabaigos požymis 0. Nustatykite ir išveskite aukščiausio ir žemiausio mokinio ūgį, bei nustatykite ar bent vienas iš visų įvestų yra didesnis už 2 m. Jeigu yra didesnių, suskaičiuokite ir išveskite kiek jų, priešingu atveju išveskite „Nėra didesnių už 2 m“

Kaupiamasis4.2.WHILE *Skaičiai ir skaitmenys*

Pavyzdinė programa3 **Skaicius.** ***Parašykite programą, kuri suskaičiuotų kiek skaitmenų yra duotame skaičiuje, bei rastų tų skaitmenų sumą ir nelygių nuliui skaitmenų sandaugą.***

```
int sk, kiek, sktmsuma, sktmsandg, sktm, skkop;

cout<<"Koks skaičius? ";
cin>>sk;
skkop=sk;
kiek=0;
sktmsuma=0;
sktmsandg=1;

while (skkop>0)
{
    sktm=skkop%10;  //atskiriamas paskutinis skaitmuo
    kiek=kiek+1;
    sktmsuma=sktmsuma+sktm;
    if (sktm != 0) sktmsandg=sktmsandg*sktm;
    skkop=skkop/10;  //sumažinamas skaičiaus skaitmenų kiekis
}
cout<<"Duotas skaičiaus "<<sk<<endl;
cout<<" skačių sudado "<<kiek<<" skaitmenys(-ų)"<<endl;
cout<<" skaitmenų suma "<<sktmsuma<<endl;
cout<<" nelygiu nuliui skaitmenų sandauga "<<sktmsandg<<endl;
```

```
Koks skaičius? 2305
Duotas skaičiaus 2305
 skačių sudado 4 skaitmenys(-ų)
 skaitmenų suma 10
 nelygiu nuliui skaitmenų sandauga 30
```

Uždavinių sprendimas įgūdžiams formuoti_2

7. *LygNelyg*

Parašykite programą, kuri suskaičiuotų, kiek duotas skaičius a turi lyginių ir kiek nelyginių skaitmenų.

Pasitikrinkite. Jei a = 63258, tai turi būti išvesta: skaičius 63258 turi 3 lyginius ir 2 nelyginius skaitmenis.

8. *MinMax*

Duotas natūralusis skaičius n. Parašykite programą, kuri rastų didžiausią ir mažiausią šio skaičiaus skaitmenį.

Pasitikrinkite. Jei n = 4328, tai turi būti išvesta: mažiausias skaitmuo 2, didžiausias 8.

9. *Didesni*

Duotas natūralusis skaičius n. Parašykite programą, kuri nustatytų kiek jame yra skaitmenų nemažesnių už duotą skaitmenį k.

Pasitikrinkite. Jei n = 63258, k=5, tai turi būti išvesta: Skaičiuje 63258 yra 3 skaitmenys nemažesni už 5

10. *Polindromas*

Duotas natūralusis skaičius n. Parašykite programą, kuri nustatytų ar jis yra polindromas (skaitomas iš abiejų pusių vienodai)

Pasitikrinkite. Jei n = 4334, tai turi būti išvesta: TAIP, duotas skaičius yra polindromas

11. *Trecdalis*

Duotas natūralusis skaičius n. Parašykite programą, kuri apskaičiuotų jo nelyginių skaitmenų sumos trečdalį (⅓). Rezultatą išvesti dviejų skaitmenų po kablelio tikslumu.

Pasitikrinkite. Jei n = 279, tai turi būti išvesta: 5.33

Kaupiamasis4.2.WHILE *Realus turinys*

Pavyzdinė programa4 ***Martynas labai mėgsta saldainius. Mamos slėptuvėje berniukas rado m saldainių. Pirmą dieną Martynas suvalgė 1 saldainį, antrą – 2, trečią – 3 ir t.t. Kiekvieną kitą dieną jis suvalgydavo vienu***

saldainiu daugiau negu prieš tai buvusią dieną. Per kelias dienas d Martynas suvalgys visus saldainius. Paskutinei dienai gali likti mažiau saldainių.

```
// Smaližius
#include <iostream>
using namespace std;

int main ()
{

 int m;    // Saldainių skaičius slėptuvėje
 suvalg;   // Kiekvieną dieną suvalgoma saldainių skaičius
 d;        // Dienų, per kurias bus suvalgyti visi saldainiai, skaičius

 cout << "Įveskite saldainių skaičių slėptuvėje: ";
 cin >> m;

 d = 0; suvalg = 0;

 while (m > 0)           // valgys kol bus saldainių
  {
    d = d + 1;                  //dienos
    suvalg = suvalg + 1;        // d dieną suvalgyti saldainiai
    m = m - suvalg;             //liko saldainių
  }
 cout << "Martynas visus saldainius suvalgys per " << d << " dienas (-ų)." << endl;

 return 0;
}
```

Pavyzdinė programa5 ***Fibonačio seka, tai skaičių eilė, kurioje kiekvienas skaičius (nuo trečiojo) yra lygus dviejų prieš jį einančių skaičių sumai. Pavyzdys: 0, 1, 1, 2, 3, 5, 8, 13...***

```
int main() {
    int n; cout << "Kiek nariu norite matyti? "; cin >> n;
    if (n > 0) {
        long long pirmas = 0, antras = 1, sekantis; // Sukuriame kintamuosius pi
        cout << "Seka: ";
        if (n >= 1) cout << pirmas << " ";
        if (n >= 2) cout << antras << " ";
        int i = 3;
        while (i <= n) {
            sekantis = pirmas + antras;
            cout << sekantis << " "; // Isvedame gauta nauja sekos nari i ekrana
            // Paslenkame skaičius į priekį būsimam žingsniui
            pirmas = antras;   // Antrasis tampa pirmuoju
            antras = sekantis; // Naujas sekantis tampa antruoju
            i++;
        }
        cout << endl;
    }
    return 0;
}
```

Uždavinių sprendimas įgūdžiams formuoti_3

12. *Slidininkas*

Slidininkas treniruotis pradėjo nušliuždamas a km nuotolį. Kiekvieną kitą treniruotę jis įveikdavo b km daugiau. Parašykite programą, nurodančią kurią, treniruotę nušliuožtas nuotolis viršys c km.

Pasitikrinkite. Jei a = 1, b = 1, c = 10, tai turi būti išvesta: 9. Jei a = 2, b = 3, c = 10, tai turi būti išvesta 3.

Pagalba

13. Bankas

Bankas už indėlius moka p procentų palūkanų per metus. Metų gale palūkanos pridedamos prie indėlio. Jei indėlininkas pinigų nė kiek neišima, palūkanos skaičiuojamos nuo vis didesnės sumos. Parašykite programą, kuri apskaičiuotų, per kiek metų t pradinis indėlis ind pasieks sumą s.

Pasitikrinkite. Kai p = 5, ind = 1000, s = 1200, turėtumėte gauti t = 4.

14. Teatras

Studentas nusprendė kiekvieną sekmadienį eiti į teatrą. Norėdamas užsidirbti tam pinigų, nešiojo rytais laikraščius ir taip gaudavo po 5 litus kiekvieną savaitės dieną. Sekmadienį studentas pirkdavo bilietą į teatrą už 10 litų, o likusius per savaitę uždirbtus pinigus išleisdavo teatro kavinėje. Tačiau bilietai pradėjo brangti. Parašykite programą, kuri apskaičiuotų, po kiek savaičių studentui nebeužteks pinigų net bilietui, jei bilietai brangsta kas dvylika savaičių po p procentų, o studento uždarbis nekinta.

Pasitikrinkite. Jei p = 50, tai pinigų bilietui nebeužteks po 48 savaičių.

15. Zeme

Gobšus žemių savininkas pradėjo supirkinėti žemę. Pirmojo jo pirkto sklypo plotas – n hektarų, antrojo – m hektarų. Toliau žemvaldžio apetitas pradėjo augti. Trečiojo sklypo plotas buvo lygus pirmojo ir antrojo sklypų plotų sumai, ketvirtojo plotas – antrojo ir trečiojo plotų sumai ir t.t. Sudarykite programą, nustatančią, kelinto gobšuolio sklypo plotas viršys s hektarų.

Pasitikrinkite. Kai n = 5, m = 6, s = 12, tai turi būti išvesta 4. Kai n = 1, m = 1, s = 10, tai turi būti išvesta 7. Kai n = 8, m = 10, s = 25, tai turi būti išvesta 4.

16. Algos

Klaviatūra renkama skaičių seka - darbuotojų atlyginimai. Įvedimas baigiamas įvedus nulį. Nustatykit didžiausią ir mažiausią gautą atlyginimą

Pasitikrinkite. Jei turime seką 1005.5, 1200.8, 2050.88, tai didžiausias 2050.88, o mažiausias - 1005.5

17. Riesutai

Jonukas ir Petriukas riešutavo. Surinkus n riešutų jie nutarė pasidalyti taip: Jonukas paims vieną riešutą, Petriukas – 2, Jonukas – 3, Petriukas – 4 ir t.t., kol riešutų bus. Paskutinis paims likusius riešutus. Sudarykite programą, apskaičiuojančią, po kiek riešutų gaus kiekvienas vaikas.

Pasitikrinkite. Jei n = 10, tai turi būti išvesta: Jonukas gaus 4, o Petriukas gaus 6. Jei n = 11, tai turi būti išvesta: Jonukas gaus 5, o Petriukas gaus 6. Jei n = 15, tai turi būti išvesta: Jonukas gaus 9, o Petriukas gaus 6. Jei n = 16, tai turi būti išvesta: Jonukas gaus 9, o Petriukas gaus 7. Jei n = 18, tai turi būti išvesta: Jonukas gaus 9, o Petriukas gaus 9.

Papildomi uždaviniai

1. Programuotojas. Programuotojas pirmą dieną išsprendė *p* uždavinių, o kiekvieną kitą dieną išspręsdavo *k* uždavinių daugiau, negu prieš tai buvusią. Parašykite programą, skaičiuojančią, per kelias dienas *d* programuotojas išspręs *u* uždavinių. Paskutinei dienai gali likti mažiau uždavinių.
Pasitikrinkite: kai p = 2, k = 1, u = 10, tuomet d = 4.

2. Lietus. Pirmąją pavasario dieną lijo *m* minučių, o kiekvieną kitą dieną *k* minučių ilgiau, negu prieš tai buvusią. Per minutę prilyja vidutiniškai *mm* milimetrų kritulių. Parašykite programą, skaičiuojančią, po kelių dienų *d* bus prilyta *s* milimetrų kritulių.
Pasitikrinkite: kai m = 5, k = 3, mm = 2, s = 30, tuomet d = 3.

3. Riešutai. Krūvelėje yra *n* riešutų. Riešutus reikia perkelti į kitą krūvelę pagal tokį algoritmą: jei krūvelėje riešutų skaičius nelyginis, tuomet į kitą krūvelę perkeliamas vienas riešutas, jei lyginis – pusė visų krūvelėje esančių riešutų. Veiksmai kartojami tol, kol visi riešutai iš vienos krūvelės 78 perkeliami į kitą. Parašykite programą, skaičiuojančią kelių perkėlimų *k* reikės, norint riešutus perkelti iš vienos krūvelės į kitą.

Pasitikrinkite: kai n = 7, k = 5.

4. Žvejyba. Užkietėjęs žvejys Kazimieras pirmą kartą šį pavasarį išsiruošė į žvejybą. Jis prigaudė ešerių. Parašykite programą, kuri apskaičiuotų: 1) bendrą sugautų žuvų masę *m*; 2) žuvų, sunkesnių kaip 100 gramų, kiekį *k*. Kiekvieno ešerio masė *me* (gramais) įvedama atskirai. Žuvų masių įvedimas baigiamas įvedus nulį.
Pasitikrinkite: kai me = 150, me = 75, me = 80, me = 0, tuomet m = 305, k = 1.

5. Saulėtos dienos. Meteorologijos stotis registruoja, kelias minutes *md* per dieną buvo giedra. Kiekvienos dienos duomenys įvedami atskirai, įvedimas baigiamas įvedus nulį. Parašykite programą, skaičiuojančią: 1) kelias dienas *d* buvo registruojama, kiek laiko per dieną buvo giedra; 2) kiek minučių *m* iš viso buvo giedra per *d* dienų; 3) kelias minutes vidutiniškai *mvid* buvo giedra per dieną.
Pasitikrinkite: kai md = 150, md = 75, md = 80, md = 0, tuomet d = 3, m = 305, mvid = 102.

6. Užsakymai. Austėja užsakinėja knygas internetiniame knygyne. Užsakymams gali skirti *s* eurų. Ji pasirenka norimą knygą, deda ją į krepšelį, žiūri, kiek liko pinigų, tuomet ieško kitos knygos ir taip daro tol, kol pinigų užtenka norimai knygai įsigyti. Parenkite programą, skaičiuojančią: 1) kelias knygas *k* įsigis Austėja; 2) kokią pinigų sumą *ks* kainuos Austėjos įsigytos knygos; 3) kiek pinigų *p* liks Austėjai, jei knygos kainuos ne lygiai tiek, kiek ji turi pinigų.
Pasitikrinkite: kai s = 25, ir knygos kainuoja: kk = 15, kk =12, tuomet k = 1, ks = 15, p = 10

Kaupiamasis4.2 Ciklas WHILE			
1pvz	*SekosSuma*	*1*	
2pvz	*KiekSkaiciu*	*1*	
1	**NeigSuma**	**1**	
2	**KiekSskaiciu**	**1**	
3	**Vidurkis**	**2**	
4	**Lyginumas**	**2**	
5	**Intervalas**	**3**	
6	**Ugiai**	**4**	
3pvz	*KiekSkaitmenu*	*1*	
7	**LygNelyg**	**1**	
8	**MinMax**	**2**	
9	**Didesni**	**3**	
10	**Polindromas**	**3**	
11	**Trecdalis**	**3**	
4pvz	*Smalizius*	*1*	
12	**Slidininkas**	**2**	
13	**Bankas**	**2**	
14	**Teatras**	**3**	
15	**Zeme**	**3**	
16	**Algos**	**2**	
17	**Riesutai**	**3**	
Vertinimas (taškai)		**25**	
	Numatomas atsiskaitomasis darbas		

Pagrindiniai algoritmai:

Sumos apskaičiavimo algoritmas

Sumos apskaičiavimo algoritmas	
`Suma = 0`	
Kartoti, kol bus naujų dėmenų	
	Gauti dėmenį
	`Suma = Suma + dėmuo`
Grąžinti `Suma`	

Sandaugos apskaičiavimo algoritmas

Sándaugos apskaičiãvimo algorìtmas mažai skiriasi nuo sumos apskaičiavimo algoritmo: iš pradžių sandaugai priskiriamas vienetas, o, atliekant ciklo veiksmus, sandauga dauginama iš naujo daugiklio:

Sandaugos apskaičiavimo algoritmas	
`Sandauga = 1`	
Kartoti, kol bus naujų daugiklių	
	Gauti daugiklį
	`Sandauga = Sandauga * daugiklis`
Grąžinti `Sandauga`	

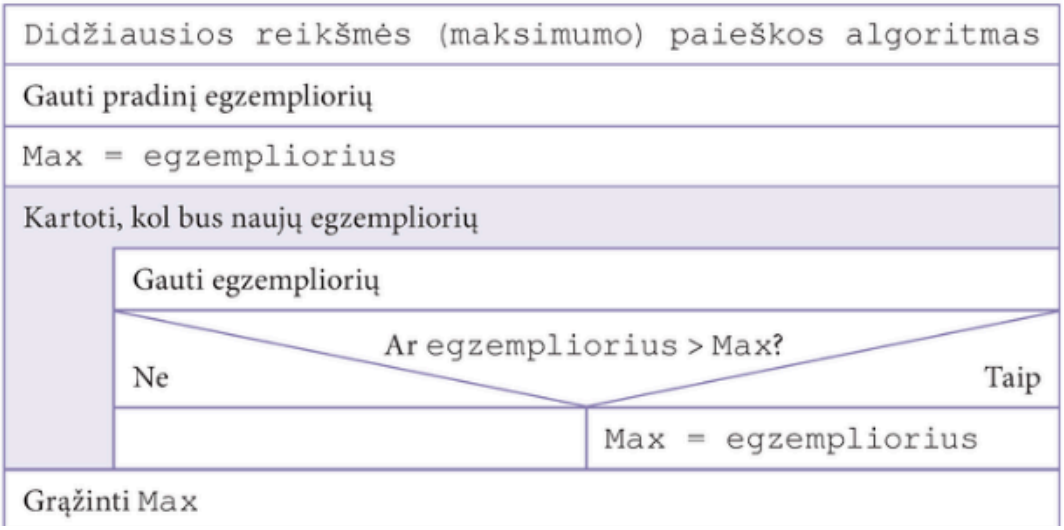
Kiekio apskaičiavimo algoritmas

Kiẽkio apskaičiãvimo algorìtmas nuo sumos apskaičiavimo algoritmo skiriasi tik tuo, kad jame sumuojami vienetai, jeigu tenkinama egzemplioriaus atrankos sąlyga:

Kiekio apskaičiavimo algoritmas	
`Kiekis = 0`	
Kartoti, kol bus naujų egzempliorių	
	Gauti egzempliorių
	Ar tinka? Ne / Taip
		`Kiekis = Kiekis + 1`
Grąžinti `Kiekis`	

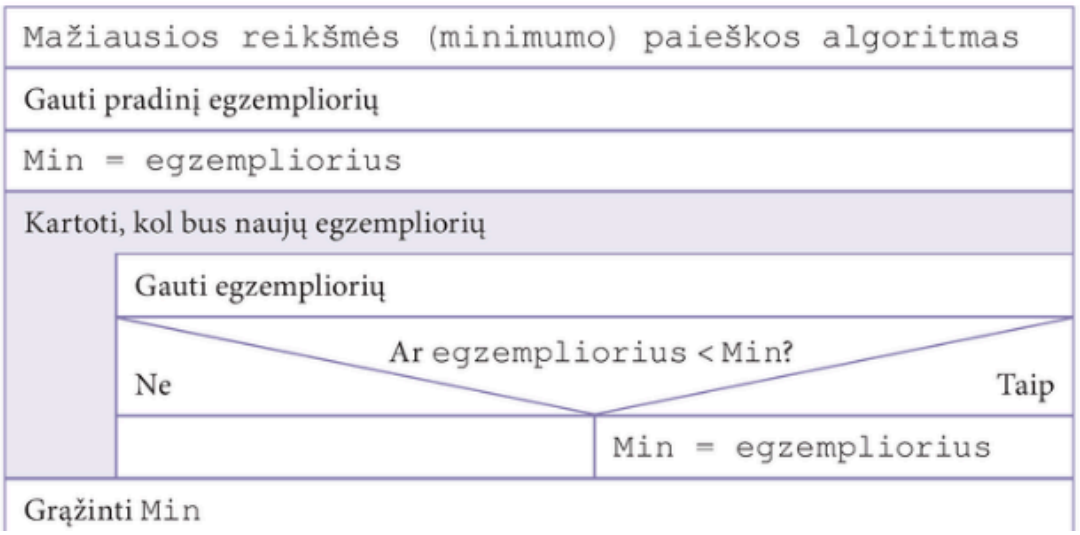
Didžiausios reikšmės (maksimumo) paieškos algoritmas

Didžiáusios reikšmẽs paieškõs algorìtmas remiasi prielaida, kad didžiausias yra pradinis egzempliorius. Toliau vykdant ciklą bandoma pagerinti rezultatą – jei randamas tinkamesnis egzempliorius, jis įsimenamas vietoj anksčiau įsimintojo.



Mažiausios reikšmės (minimumo) paieškos algoritmas

Mažiáusios reikšmẽs paieškõs algorìtmas nuo didžiausios reikšmės paieškos algoritmo skiriasi tik egzemplioriaus parinkimo sąlyga:



Aritmetinio vidurkio skaičiavimas

Aritmetinis vidurkis skaičiuojamas dviem etapais:

1) apskaičiuojama skaičių suma;
2) gauta suma padalijama iš skaičių kiekio.

Sumos skaičiavimo algoritmą jau išnagrinėjome. Reikia atkreipti dėmesį, kad jeigu iš anksto nežinoma, kiek skaičių sumuojama, tai sumuojant reikia kartu skaičiuoti, kiek tokių skaičių buvo.

Antrasis etapas – tai dviejų skaičių (gautos sumos ir skaičių kiekio) dalyba:

```
vidurkis = suma / skaičiųKiekis.
```

Žinome, kad dalyba iš nulio negalima. Todėl, prieš atliekant dalybos veiksmą (pvz., kai reikia apskaičiuoti teigiamų skaičių aritmetinį vidurkį), būtina įsitikinti, kad galima dalyti (kad įvesti ne vien neigiami skaičiai ir nuliai).