

Pavyzdinis uždavinys „Ligonis“

UŽDUOTIS

Ligonis. Ligonio temperatūra per parą matuojama kas valandą arba kas kelias valandas ir užrašoma į ligonio kortelę. Remdamasis šiais įrašais, gydytojas ligoniui skiria gydymą. Parašykite programą, kuri rastų:

- ✓ kurią valandą temperatūra buvo aukščiausia;
- ✓ kuriomis matavimo valandomis temperatūra buvo artima (± 0.5 laipsnio) aukščiausiai temperatūrai.

Pirmoje pradinių duomenų failo eilutėje įrašyta, kiek kartų per parą buvo matuota ligonio temperatūra. Likusiose eilutėse yra po du tarpais atskirtus skaičius: valanda (sveikasis skaičius), kada matuota, ir temperatūra (realusis skaičius).

Pradinių duomenų ir rezultatų failų pavyzdys

Pradiniai duomenys	Rezultatai
10	Ligonio temperatūra
5 38.5	-----
6 38.1	Valanda Temperatūra
8 38.6	-----
10 37.2	5 38.5
16 39.5	6 38.1
17 39.7	8 38.6
18 39.7	10 37.2
20 40.1	16 39.5
21 39.5	17 39.7
22 39.8	18 39.7
	20 40.1
	21 39.5
	22 39.8

	Aukščiausia temperatūra 40.1 buvo 20 val.
	Aukšta temperatūra dar buvo:
	17 val. 39.7
	18 val. 39.7
	20 val. 40.1
	22 val. 39.8

Algoritmas

Algoritmas

Spręsdami užduotį, naudosisimės vienu sveikųjų ir vienu realiųjų skaičių masyvu.

Užduotis gali būti sprendžiama taip:

1. Duomenys – valandos ir temperatūros tomis valandomis – skaitomi iš failo į du masyvus.
2. Masyvų elementų reikšmės rašomos į rezultatų failą lentelė: valanda, temperatūra.
3. Atliekami skaičiavimai:
 - ✓ kurią valandą temperatūra buvo aukščiausia;
 - ✓ kuriomis matavimo valandomis temperatūra buvo artima (± 0.5 laipsnio) aukščiausiai temperatūrai.
4. Apskaičiuoti rezultatai rašomi į rezultatų failą.

1) Skaitymo procedūra [c++]

```
1  #include <iostream>
2  #include <fstream>
3  #include <iomanip>
4  #include <cmath>
5  using namespace std;
6
7  void Skaityti(int A[], int & n, double B[])
8  {
9      ifstream fin("Duomenys.txt");
10     fin >> n;
11     for(int i = 1; i <= n; i++)
12     {
13         fin >> A[i] >> B[i];
14     }
15
16     fin.close();
17 }
```

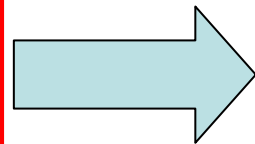
2) Pagrindinė programa: Skaitymas ir Spausdinimas [c++]

```
48 int main()
49 {
50     int A[1000];
51     double B[1000];
52     int n;
53     ofstream fout("Rezultatai.txt");
54     Skaityti(A, n, B);
55     fout << " Ligonio  temperatura  \n";
56     fout << " ----- \n";
57     fout << " Valanda  Temperatura  \n";
58     for(int i = 1; i <= n; i++)
59         fout << setw(5) << A[i] << " " << fixed << setw(2) << setprecision(1) << B[i] << endl;
60     fout << " ----- \n";
```

Duomenų ir rezultatų failai

Duomenys

```
10
5 38.5
6 38.1
8 38.6
10 37.2
16 39.5
17 39.7
18 39.7
20 40.1
21 39.5
22 39.8
```



P
R
O
G
R
A
M
A



Rezultatai

```
Ligonio temperatūra:
-----
Valanda  Temperatūra
-----
5        38.5
6        38.1
8        38.6
10       37.2
16       39.5
17       39.7
18       39.7
20       40.1
21       39.5
22       39.8
-----
```

Funkcija Auksciausia(), procedūra Auksta() [C++]

```
18  int Auksciausia(int n, double B[])
19  {
20      int ind = 1;
21      for(int i = 1; i <= n; i++)
22      {
23          if (B[i] > B[ind])
24              ind = i;
25      }
26      return ind;
27  }
28  void Auksta(double B[], int n, int A[])
29  {
30      ofstream fout("Rezultatai.txt", ios::app);
31      double max = B[Auksciausia(n, B)];
32      fout << " Auksciausia temperatura dar buvo: \n";
33      for(int i = 1; i <= n; i++)
34      {
35          if (max - B[i] <= 0.5)
36              fout << " " << A[i] << " val. " << B[i] << endl;
37      }
38      fout.close();
39  }
```

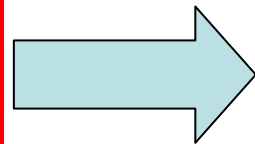
Programa [C++]

```
42 int main()
43 {
44     int A[1000];
45     double B[1000];
46     int n;
47     ofstream fout("Rezultatai.txt");
48     Skaityti(A, n, B);
49     fout << " Ligonio temperatura \n";
50     fout << " ----- \n";
51     fout << " Valanda Temperatura \n";
52     for(int i = 1; i <= n; i++)
53         fout << setw(5) << A[i] << " " <<
54             fixed << setw(2) << setprecision(1) << B[i] << endl;
55     fout << " ----- \n";
56     fout << " Didziausia temperatura " << B[Auksciausia(n, B)] <<
57         | " buvo " << A[Auksciausia(n, B)] << " val. " << endl;
58     Auksta(B, n, A);
59
60     fout.close();
61     return 0;
62 }
```

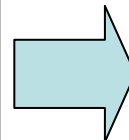

Duomenų ir rezultatų failai

Duomenys

```
10
5 38.5
6 38.1
8 38.6
10 37.2
16 39.5
17 39.7
18 39.7
20 40.1
21 39.5
22 39.8
```



P
R
O
G
R
A
M
A



Rezultatai

```
Ligonio temperatūra:
-----
Valanda  Temperatūra
-----
5          38.5
6          38.1
8          38.6
10         37.2
16         39.5
17         39.7
18         39.7
20         40.1
21         39.5
22         39.8
-----
Aukščiausia temperatūra 40.1 buvo 20 val.
Aukšta temperatūra dar buvo:
17 val. 39.7
18 val. 39.7
20 val. 40.1
22 val. 39.8
```

Uždaviniai

1. Grybai

Petriukas labai mėgsta grybauti. Kiekvieną kartą, sugrįžęs iš miško, savo dienoraštyje jis užsirašo visų surinktų grybų masę ir rastų baravykų bei raudonviršių skaičių. Parenkite programą ir sužinokite, kuri grybavimo sezono diena buvo derlingiausia, kurią dieną Petriukas surinko daugiausia baravykų ir kurią – daugiausia raudonviršių.

Pirmoje pradinių duomenų failo eilutėje nurodytas grybavimo dienų skaičius n ($1 \leq n \leq 100$). Kitose n eilučių yra po penkis skaičius: mėnuo, diena, parneštų grybų masė, baravykų skaičius ir raudonviršių skaičius.

Pradiniai duomenys	Rezultatai
5 7 4 12.5 5 0 7 15 14.2 0 0 7 16 5.5 8 1 8 13 10.1 15 25 8 25 3.5 0 4	Derlingiausia diena: 7 15 Derlingiausia baravykų diena: 8 13 Derlingiausia raudonviršių diena: 8 13
2 7 4 12.5 5 0 8 25 3.5 0 0	Derlingiausia diena: 7 4 Derlingiausia baravykų diena: 7 4 Derlingiausia raudonviršių diena: nėra

Uždaviniai

2. Žibintai

Parką apšviečia n ($1 \leq n \leq 100$) žibintų. Kiekvienas jų visą naktį šviečia arba ne. Žinoma, kurie žibintai pirmą naktį švietė, o kurie ne. Kiekvieną kitą naktį žibintų būseną nusako tokios taisyklės:

- ✓ žibintas nešvies, jeigu praeitą naktį abu jo kaimynai švietė;
- ✓ žibintas švies, jeigu praeitą naktį vienas jo kaimynas švietė, o kitas nešvietė;
- ✓ jeigu žibintui negalioja nė viena pirmų dviejų taisyklių, tai jis švies, kai naktis lyginė, ir nešvies, kai naktis nelyginė.

Parenkite programą, kuri nustatytų, kurie žibintai švies ir kurie nešvies, kai praeis k ($1 \leq k \leq 100$) naktų, ir rastų, kurią naktį švietė daugiausia žibintų ir kurią – mažiausiai. Jeigu yra kelios vienodos naktys, kai švietė daugiausia ar mažiausiai žibintų, tai reikia nurodyti tą naktį, kurios numeris mažesnis.

Pirmoje pradinių duomenų failo eilutėje yra du sveikieji skaičiai: žibintų skaičius n ir naktų skaičius k . Antrą eilutę yra nulių ir vienetų, atskirtų vienu tarpu, seka. Tai kiekvieno žibinto būseną: vienetą reiškia, kad pirmą naktį žibintas švietė, o nulis, kad nešvietė.

Pradiniai duomenys	Rezultatai	Paiškinimas
6 2 0 1 0 1 1 1	1 1 0 1 0 1 Naktis, kai švietė daugiausia: 1 Naktis, kai švietė mažiausiai: 1	0 1 0 1 1 1 Pirma naktis 1 1 0 1 0 1 Antra naktis
9 5 0 1 1 1 1 0 1 0 1	0 1 0 1 0 0 0 0 0 Naktis, kai švietė daugiausia: 4 Naktis, kai švietė mažiausiai: 5	0 1 1 1 1 0 1 0 1 Pirma naktis 1 1 0 0 1 0 1 0 1 Antra naktis 0 1 1 1 0 0 0 0 0 Trečia naktis 1 1 0 1 1 1 1 1 1 Ketvirta naktis 0 1 0 1 0 0 0 0 0 Penkta naktis

Uždaviniai

3. Dalelė

Mokslininkas stebi vandens paviršiuje pakibusios kietosios dalelės judėjimą ir kas n sekundžių užrašo jos koordinates. Parenkite programą, kuri apskaičiuotų kietosios dalelės mažiausią, vidutinį bei didžiausią greičius ir dalelės nukeliautą atstumą.

Pirmoje pradinių duomenų failo eilutėje įrašytas dalelės koordinačių skaičius m ($1 < m \leq 100$) ir koordinačių fiksavimo intervalas n ($n > 0$) sekundėmis. Tolesnėse eilutėse nurodytos dalelės koordinatės x_i ir y_i ($1 \leq i \leq m$) milimetrais (x_i ir y_i – realieji skaičiai).

Į rezultatų failą įrašykite kietosios dalelės mažiausią, vidutinį bei didžiausią greičius ir dalelės nukeliautą atstumą.

Pradiniai duomenys	Rezultatai
5 2 1.8 2.7 -0.5 4.3 -1.4 3.2 0.5 1.4 0.7 -0.5	Mažiausias greitis: 0.71 mm/s Vidutinis greitis: 1.09 mm/s Didžiausias greitis: 1.40 mm/s Nukeliautas atstumas: 8.76 mm