1.2 Darbas su failais. Duomenų skaitymas ir rašymas.

# Užduotys

## 1.Uždarbis

Per pirmą darbo mėnesį žmogus uždirbo ***p1*** eurų, per antrąjį - ***p2*** ir t.t.

Parenkite programa, kuri apskaičiuotų, kiek pinigų iš viso uždirbo žmogus per ***n*** mėnesių.

Pirmoje pradinių duomenų failo eilutėje įrašytas mėnesių skaičius ***n***. Tolesnėse n eilučių įrašyta po vieną realųjį skaičių - žmogaus mėnesio atlyginimas.

Rezultatų faile turi būti įrašyta žmogaus uždirba per ***n*** mėnesių pinigų suma.

| Pradiniai duomenys | Rezultatas |
| --- | --- |
| 3  700.50  600.25  600.25 | 1901.00 |

Programa:

| #include <iostream>  #include <fstream>  #include <iomanip>  using namespace std;  int main()  {  float suma = 0;  float menesiai;  float kintamasis;  ifstream fin("Duomenys.txt");  ofstream fout("Rezultatas.txt");  fin >> menesiai;  for (int i = 0; i < menesiai; i++)  {  fin >> kintamasis;  suma += kintamasis;  }  fout << fixed << setprecision(2) << suma;  return 0;  } |
| --- |
|  |

## 

## 2.Prekyba ledais

Pirmąją prekybos dieną verslininkas pardavė ***k*** porcijų ledų.Prekyba sekėsi gerai - kiekvieną kitą dieną jis parduodavo po m porcijų ledų daugiau negu prieš tai buvusią.

Parenkite programą, kuri apskaičiuotų, kiek porcijų ledų ***vk*** verslininkas pardavė per ***n*** dienų.

Pradinių duomenų faile įrašyti trys atskirti tarpais skaičiai: ***k*** (kiek pirmąją prekybos dieną parduota porcijų), ***m*** (keliomis porcijomis kiekvieną kitą dieną buvo parduodama daugiau) ir ***n*** (kelias dienas buvo prekiauta ledais).

Rezultatų faile turi būti įrašyta, kiek porcijų ledų verslininkas pardavė per ***n*** dienų.

| Pradiniai duomenys | Rezultatas |
| --- | --- |
| 170 30 3 | 600 |

Programa:

| #include <iostream>  #include <fstream>  using namespace std;  int main()  {  int pirma\_diena;  int daugejimas;  int dienos;  ifstream fin("Duomenys.txt");  ofstream fout("Rezultatas.txt");  fin >> pirma\_diena;  fin >> daugejimas;  fin >> dienos;  int viso = pirma\_diena;  for (int i = 1; i < dienos; i++)  {  pirma\_diena += daugejimas;  viso += pirma\_diena;  }  fout << viso;  return 0;  } |
| --- |

## 3.Kačių dresuotojas

Pramuštgalvis penkiametis Audrius nusprendė dresuoti kates. Audrius mokosi dresuodamas savo katiną Kiviną. Pirmą dresavimo dieną Kivinas buvo dresuojamas ***t1*** minučių, antrąją - ***t2*** ir t. t.

Parenkite programa, kuri apskaičiuotų, kiek minučių buvo dresuojamas Kivinas, jei Audrius jį drėsavo ***n*** dienų, ir kiek vidutiniškai minučių Kivinas buvo dresuojamas per dieną.

Pirmuoje pradinių duomenų failo eilutėjė įrašytas dienų skačius n. Tolesnėse ***n*** eilučių įrašyta po vieną sveikąjį skaičių - kiek minučių buvo dresuojamas katinas kiekvieną dieną.

Rezultatų faile turi būti įrašyta, kiek minučių buvo dresuojamas katinas per ***n*** dienų ir kiek vidutiniškai minučių Kivinas buvo dresuojamas per dieną.

| Pradiniai duomenys | Rezultatas |
| --- | --- |
| 4  15  12  13  20 | 60  15 |

Programa:

| #include <iostream>  #include <fstream>  using namespace std;  int main()  {  int dienos;  int kintamasis;  int viso = 0;  ifstream fin("Duomenys.txt");  ofstream fout("Rezultatas.txt");  fin >> dienos;  for (int i = 0; i < dienos; i++)  {  fin >> kintamasis;  viso += kintamasis;  }  fout << viso << endl << viso / dienos;  return 0;  } |
| --- |

## 4.Biatlonininko rezultatas

Biatlono varžybose yra ***n*** etapų, kuriuose reikia po ***x*** kartų pataikyti į taikinį. Pirmą etapą varžybų dalyvis įveikė per ***t1*** minučių ir pataikė ***k1*** kartų, antrąjį - per ***t2*** minučių ir pataikė ***k2*** kartų ir t.t. Už kiekvieną nepataikytą kartą sportininkui pridedama ***b*** baudos minučių.

Parenkite programą, kuri apskaičiuotų, per kiek minučių varžybų dalyvis įveikė trasą. Spręsdami uždavinį, turėkite omenyje, kad pradiniai duomenys ir rezultatai yra sveikieji skaičiai.

Pirmoje pradinių duomenų failo eilutėje įrašytas dienų skaičius ***n***, šūvių skaičius ***x*** ir baudos minučių skaičius ***b***. Skaičiai vienas nuo kito atskirti tarpais. Tolesnėse ***n*** eilučių įrašyta po du tarpais atskirtus skaičius:

laikas, per kurį dalyvis įveikė varžybų etapą, ir kiek kartų jis pataikė į taikinį.

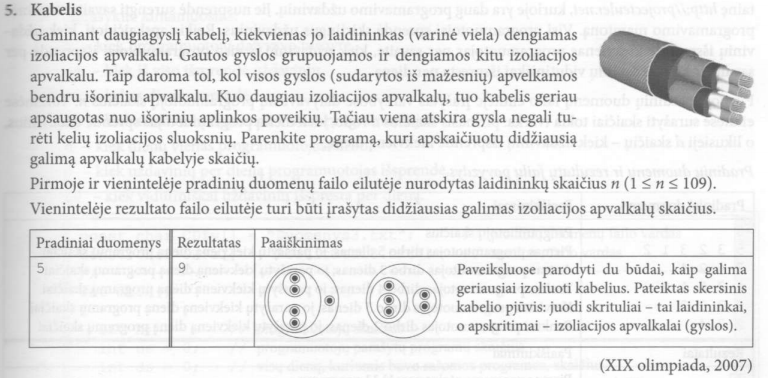
Rezultatų failė turi būti įrašyta, per kiek minučių varžybų dalyvis įveikė trasą.

| Pradiniai duomenys | Rezultatas |
| --- | --- |
| 4 5 3  5 3  3 2  3 1  5 4 | 46 |

Programa:

| #include <iostream>  #include <fstream>  using namespace std;  int main()  {  int etapai;  int taikiniai;  int bauda;  int laikas = 0;  int kintamasis;  ifstream fin("Duomenys.txt");  ofstream fout("Rezultatas.txt");  fin >> etapai;  fin >> taikiniai;  fin >> bauda;  for (int i = 0; i < etapai; i++)  {  fin >> kintamasis;  laikas += kintamasis;  fin >> kintamasis;  laikas += bauda \* (taikiniai - kintamasis);  }  fout << laikas;  return 0;  } |
| --- |
|  |

## 5.Kabelis



Gaminant daugiagyslį kabelį, kiekvienas jo laidas (varinė viela) dengiamas izoliacijos apvalkalu.Gautos gyslos grupuojamos ir dengiamos kitų izoliacijos apvalkalu. Taip daroma tol, kol visos gyslos (sudarytos iš mažesnių) apvelkamos bendru išoriniu apvalkalu. Kuo daugiau izoliacijos apvalkalų tuo kabelis geriau apsaugotas nuo išorinių aplinkos poveikių. Tačiau viena atskira gysla negali turėti kalių izoliacijos sluoksnių.

Parenkite programą, kuri apskaičiuotų didžiausią galimą apvalkalų kabelyje skaičių.

Pirmoje ir vienintelėje pradinių duomenų failo eilutėje nurodytas laidininkų skaičius ***n*** (1 <= ***n*** <= 109).

Vienintelėje rezultato failo eilutėje turi būti įrašytas didžiausias galimas izoliacijos apvalkalu skaičius.

| Pradiniai duomenys | Rezultatas | Paaiškinimas | | |
| --- | --- | --- | --- | --- |
| 5 | 9 |  |  | Paveiksluose parodyti du būdai, kaip galima geriausiai izoliuoti kabelius. Pateiktas skersinis kabelio pjūvis: juodi skrituliai - tai laidininkai, o apskritimai - izoliacijos apvalkalas (gyslos). |

Programa:

| #include <iostream>  #include <fstream>  using namespace std;  int main()  {  ifstream fin("Duomenys.txt");  ofstream fout("Rezultatas.txt");  int kabeliai; fin >> kabeliai;  fout << 2 \* kabeliai - 1;  return 0;  } |
| --- |

## 

## 6.Superfibonačio skaičiai

Tikriausiai esate girdėję apie Fibonačio skaičių seką: 1 1 2 3 5 8 … .

Ji apibrėžiama tokiu būdu : F1 = 1, F2 = 1, Fn =Fn - 1 + Fn - 2, t. y. kiekvienas narys, pradedant treciuoju, lygus prieš ji esančių dviejų narių sumai.

Panašiai apibrėžkime superfibonačio skaičius: F1 = 1, F2 = 1, Fn = Fn - 1 + Fn - 2 + Fn - 3, t. y. pirmieji trys sekos nariai yra vienetai, o kiekviena tolesnis narys gaunamas sudėjus tris paskutinius sekos narius.

Parenkite programa, kuri rastų n - ąjį superfibonačio sekos narį.

pirmoje ir vienintelėje pradinių duomenų failo eilutėjė įrašytasvienas svikas skaičius n (1<= n < 20) - ieškomo superfibonačio sekos nario numeris.

Rezultatų faile turi būti vienas skaičius - n-asis superfibonačio sekos narys.

| Pradiniai duomenys | Rezultatas | Paaiškinimas |
| --- | --- | --- |
| 6 | 8 | 1, 1, 1, 3, 5, **8** |

Programa:

| #include <iostream>  #include <fstream>  using namespace std;  int main()  {  int kintamasis;  int superfibonacci[20] = {1, 1, 1, 3, 5, 9, 17, 31, 57, 105, 193, 355, 653, 1201, 2209, 4063, 7473, 13745, 25281, 46499};  ifstream fin("Duomenys.txt");  ofstream fout("Rezultatas.txt");  fin >> kintamasis;  fout << superfibonacci[kintamasis - 1];  return 0;  } |
| --- |