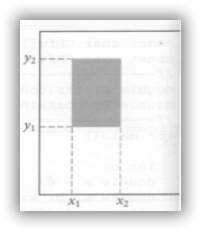
# **1.4 užduotys**

### **Reikalavimai:**

* Jeigu nenurodyta kitaip, programa duomenys gauna iš duomenų failo.
* Visose programose yra patikra ar duomenų failas egzistuoja.
* Pateikiant užduotys reikia pateikti ir duomenų failą
* Visose programose naudojamos funkcijos.
* Visose programose komentuojamas programos kodas.

### Užduotis 01

**Siena**

Tomas kolekcionuoja plakatus. Vieną dieną jis sugalvojo visus turimus plakatus suklijuoti ant savo kambario sienos. Šiek tiek pamąstęs, jis popieriaus lape sužymėjo tikslias plakatų vietas ant sienos. Tomas noi matyti visą savo kolekciją, todėl stengėsi, kad nė vienas plakatas neuždengtų kitų. Visi plakatai yra stačiakampiai ir pagal Tomo sudarytą planą jų kraštinės turi būti lygiagrečios su sienos šonais.

Tomo mama susirupinusi - neseniai dažyta siena dabar bus uždengta plakatais. Jai įdomu, kiek sienos lik neuždengta.

Parenkite programą, kuri apskaičiuotų nepaslėptos po plakatais sienos dalies plotą. Primename, kad stačiakampio plotas lygus jo kraštinių ilgių sandaugai.

Pradiniai duomenys pateikti faile.

Pirmoje failo eilutėje įrašyti du skaičiai -sienos plotis ***p*** ir aukštis ***a*** centimetrais (1 p, a 1000).

Antroje eilutėje įrašytas vienas sveikasis skaičius N (1 N 1000) - plakatų kiekis. Tolesnėse

N eilučių įrašyta po keturis sveikuosius skaičius , , , (0 < p; 0 < a): - atstumas nuo kairiojo sienos krašto iki kairiosios plakato kraštinės; - atstumas nuo apatinio sienos krašto iki apatinės plakato kraštinės; atitinkamai , ir - atstumai iki plakato dešiniosios ir viršutinės kraštinių. Visi atstumai nurodyti centimetrais.

| rezultatų failą įrašykite vieną skaičių - neuždengtos plakatais sienos dalies plotą kvadratiniais centimetrais.

| Pradiniai duomenys | Rezultatas |
| --- | --- |
| 300 200  2  50 100 150 150  200 0 300 100 | 45000 |

Programa:

| #include <iostream> // Prijungiame biblioteką, kuri yra atsakinga už išvedimo bei įvedimo funkcijas  #include <fstream> // Prijungiame biblioteką, kuri yra atsakinga už failo sukūrimą, nuskaitymą bei išvedimą informacijos jame  using namespace std; // trumpinama į std  int main()  {  ifstream input("Duomenys.txt"); // Atidarome duomenų failą  ofstream output("Rezultatai.txt"); // Atidarome rezultatų failą    if (!input){  cout << "Failas nerastas";  return 1; // IF funkcija, atsakinga už failo veikimą  }  int plotas; int x1, x2, y1, y2; int trump; // Susirašome kintamuosius - ploto, matmenų  input >> plotas;  input >> trump;  plotas \*= trump; // Ieškome ploto sienos  input >> trump;  for (int i = 0; i < trump; i++){  input >> x1 >> y1 >> x2 >> y2;  plotas -= (x2 - x1) \* (y2 - y1); // Apskaičiuojame plotį (plakatų)  }  output << plotas;  } |
| --- |

### Užduotis 02

**Dangoraižis**

Naujame dangoraižyje yra **n** kabinetų, sunumeruotų nuo 1 iki **n**. Prie kiekvieno kabineto durų reikia prikabinti lenteles su to kabineto numeriu. Ant vienos lentelės užrašytas tik vienas skaitmuo. Jei kabineto numeris vienaženklis skaičius, tai prie durų kalamos kelios lentelės.

Parenkite programą, kuri apskaičiuotų, kiek reikės lentelių, norint prikalti numerius prie visų dangoraižio kabinetų durų.

Pradinių duomenų faile įrašyta, kiek kabinetų yra dangoraižyje.

Rezultatų faile turi būti įrašytas lentelių skaičius.

| Pradinis duomuo | Rezultatas |
| --- | --- |
| 16  9999 | 23  38889 |

Programa:

| #include <iostream> // Prijungiame biblioteką, kuri yra atsakinga už išvedimo bei įvedimo funkcijas  #include <fstream> // Prijungiame biblioteką, kuri yra atsakinga už failo sukūrimą, nuskaitymą bei išvedimą informacijos jame  using namespace std; // trumpinama į std  int main()  {  ifstream input("Duomenys.txt"); // Atidarome duomenų failą  ofstream output("Rezultatai.txt"); // Atidarome rezultatų failą  if (!input){  cout << "Failas nerastas";  return 1; // IF funkcija, atsakinga už failo veikimą  }  int trump; input >> trump;  int suma = 0;  if (trump > 9){  if (trump > 99){  if (trump > 999){  suma += (trump - 999) \* 4;  trump = 999;  }  suma += (trump - 99) \* 3;  trump = 99;  }  suma += (trump - 9) \* 2; // Apskaičiuojame rezultatą  }  suma += 9;  output << suma; // Išvedame sumą (atsakymą  } |
| --- |

### Užduotis 03

**Trys sodininkai**

Trys draugai, apsigyvenę kaime, nusprendė mokytis sodininkauti. Kaime buvo didžiulis sodas, kurio kiekviename vienetiniame plotelyje augo po vieną vaismedį.

Kiekvienas iš trijų draugų pasirinko stačiakampį sklypą ir nusprendė prižiūrėti jame esančius medžius. Susirinkus draugams paaiškėjo, kad jų pasirinkti sklypai įsiterpia vienas į kitą, t. y. kai kuriuos vaismedžius prižiūrės ne vienas, o keletas sodininkų.

Parenkite programą, kuri apskaičiuotų, kiek vaismedžių panorėjo prižiūrėti visi trys draugai.

Į kiekvieną eilutę įrašyta po keturis skaičius, apibūdinančius kiekvieno draugo pasirinktą sklypą: sklypo apatinio kairiojo ir viršutinio dešiniojo kampų koordinatės (pirma koordinatė x, po to - y). Visos koordinatės - sveikieji skaičiai.

Rezultatų faile turi būti įrašytas vienas skaičius - kiek vaismedžių panorėjo prižiūrėti visi trys draugai.

| Pradiniai duomenys | Rezultatas |
| --- | --- |
| 30 30 80 70  10 20 70 90  50 20 100 90 | 800 |

Programa:

| #include <iostream> // Prijungiame biblioteką, kuri yra atsakinga už išvedimo bei įvedimo funkcijas  #include <fstream> // Prijungiame biblioteką, kuri yra atsakinga už failo sukūrimą, nuskaitymą bei išvedimą informacijos jame  using namespace std; // trumpinama į std  int main()  {  ifstream input("Duomenys.txt"); // Atidarome duomenų failą  ofstream output("Rezultatai.txt"); // Atidarome rezultatų failą  if (!input){  cout << "Failas nerastas";  return 1; // IF funkcija, atsakinga už failo veikimą  }    int xmin, xmax, ymin, ymax, trump;  input >> xmin >> ymin >> xmax >> ymax;  for (int i = 0; i < 2; i++){  input >> trump;  if (trump > xmin) xmax = trump;  input >> trump;  if (trump > ymin) ymin = trump;  input >> trump;  if (trump < xmax) xmax = trump;  input >> trump;  if (trump < ymax) ymax = trump; // Apskaičiuojame naudojantis for ir int funkcijomis  }  output << (xmax - xmin) \* (ymax - ymin); // Išvedamas atsakymas  } |
| --- |