# **1.3 Vienmačiai masyvai: informacijos įvedimas ir gavimas (išvedimas)**

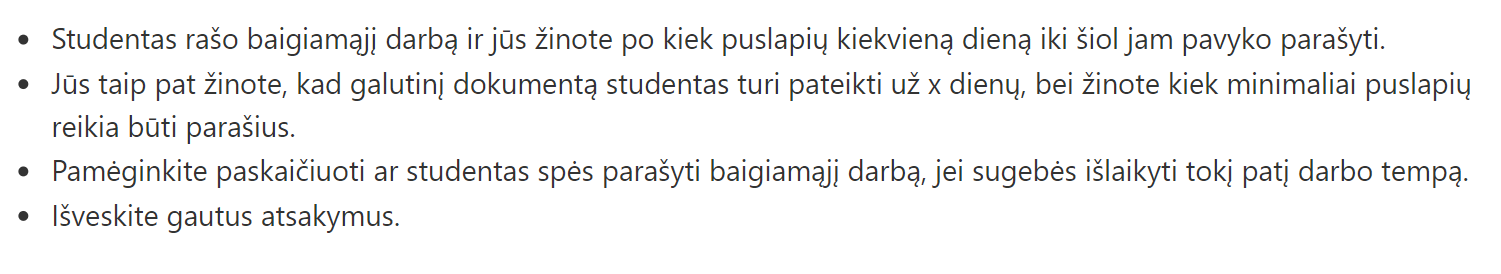
## **Užduotys 5, 6, 7**

### 

### 

### **Užduotis 5**

Sąlyga



Programa

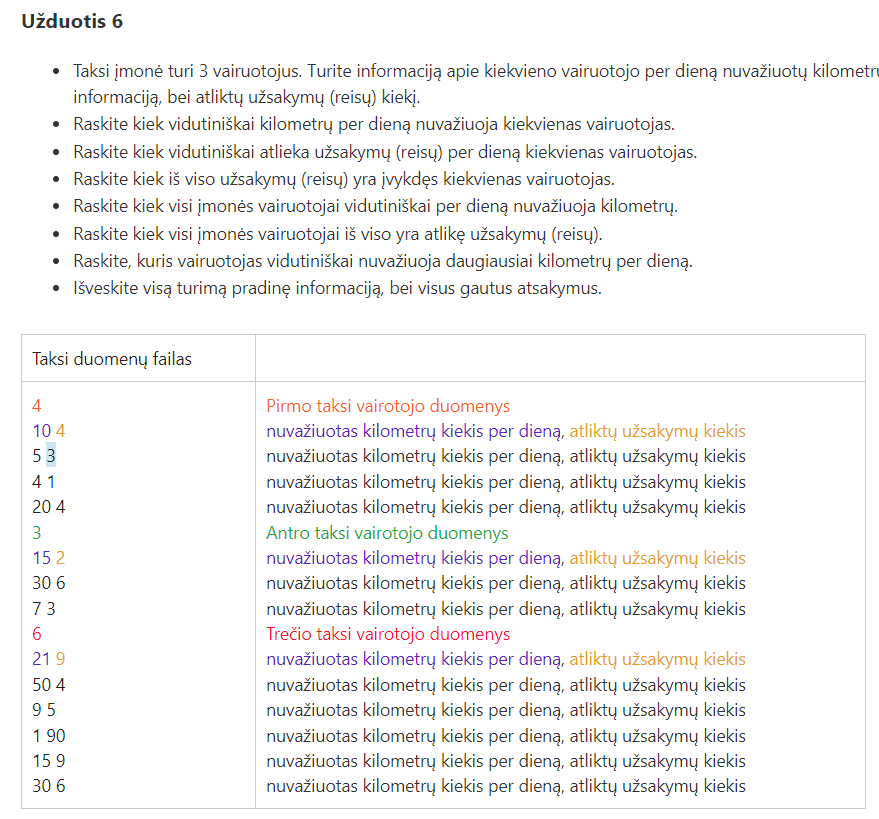
| // --- Reikalingos bibliotekos ---  #include <iostream>  #include <fstream>  using namespace std;  // --- Pagr. Funkcija ---  int main(){  //Prisijungimas prie duomenu failo  ifstream input("Duomenys.txt");  if (!input) {  cout << "Failas nerastas";  return 1;  }  //Sukuriami kintamieji  int likusios\_dienos; input >> likusios\_dienos;  int psl\_skaicius; input >> psl\_skaicius;  int dienos; input >> dienos;  float vidurkis = 0; int temp;  //suskaiciuojamas likusiu puslapiu skaicius  for (int i = 0; i < dienos; i++){  input >> temp;  vidurkis += temp;  }  //suskaiciuojama, ar studentas spes parasyti darba  if (psl\_skaicius - vidurkis - vidurkis/dienos\*likusios\_dienos <= 0)  cout << "Studentas spes parasyti baigiamaji darba";  else cout << "Studentas nespes parasyti baigiamojo darbo";  return 0;  } |
| --- |

Duomenu failas

| 6 (uz kiek dienu tures pateikti darba)  10 (kiek psl turi but parases)  3 (kiek dienu iki siol rase ir toliau per diena parasytu psl skaicius)  1  2  3 |
| --- |

### **Užduotis 6**

Sąlyga

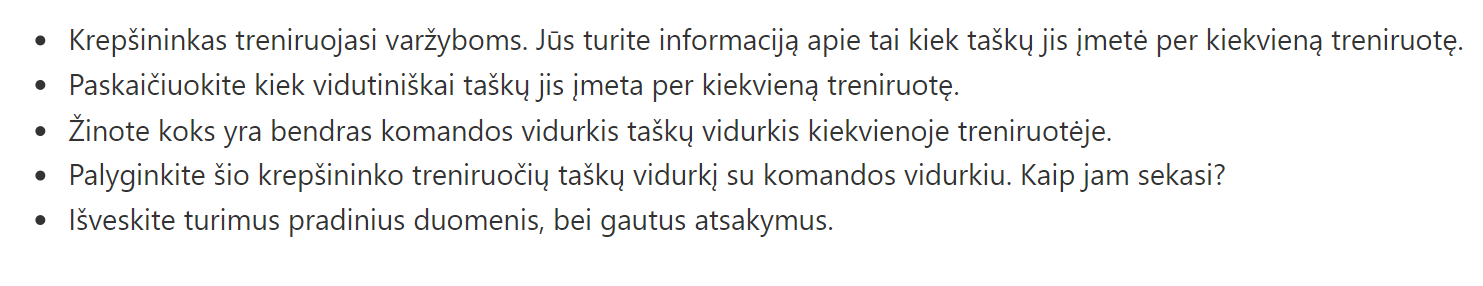


Programa

| // --- Reikalingos bibliotekos ---  #include <iostream>  #include <fstream>  using namespace std;  // --- Funkcija, apskaiciuoti individualius parametrus ---  float individualus(int array[], int kiekis){  float uzsakymai = 0;  float kilometrai = 0;  for (int i = 0; i < kiekis\*2; i+=2){  kilometrai += array[i];  uzsakymai += array[i+1];  }  cout << "\nViso uzsakymu: " << uzsakymai;  cout << "\nVidutiniskai km: " << kilometrai/kiekis;  cout << "\nVidutiniskai uzsakymu: " << uzsakymai/kiekis;  return kilometrai/kiekis;  }  // --- Pagr. Funkcija ---  int main(){  //Prisijungimas prie duomenu failo  ifstream input("Duomenys.txt");=  if(!input){  cout << "Failas nerastas";  return 1;  }  int dienos;  //Kiekvieno vairuotojo duomenu sudejimas i masyvus ir funkcijos iskvietimas  input >> dienos; int pirmas[dienos\*2];  for(int i = 0; i < dienos\*2; i++) input >> pirmas[i];  cout << "\n\nPirmas vairuotojas: "; float greitis1 = individualus(pirmas, dienos);  input >> dienos; int antras[dienos\*2];  for(int i = 0; i < dienos\*2; i++) input >> antras[i];  cout << "\n\nAntras vairuotojas: "; float greitis2 = individualus(antras, dienos);  input >> dienos; int trecias[dienos\*2];  for(int i = 0; i < dienos\*2; i++) input >> trecias[i];  cout << "\n\nTrecias vairuotojas: "; float greitis3 = individualus(trecias, dienos);  //Skaiciuojamos bendros reiksmes  input.close();  input.open("Duomenys.txt");  int viso\_uzsakymu = 0;  cout << "\n\nPradiniai duomenys:\n";  for (int i = 0; i < 3; i++){  input >> dienos; int temp;  cout << dienos << endl;  for (int k = 0; k < dienos\*2; k+=2){  input >> temp; cout << temp << " ";  input >> temp; cout << temp << "\n";  viso\_uzsakymu+=temp;  }  }  //Skaiciuojamas viso uzsakymu ir visu vidutini km skaiciu  cout << "\nViso uzsakymu: " << viso\_uzsakymu;  cout << "\nVisu vidutinis km skaicius: " << (greitis1+greitis2+greitis3)/3;  //Skaiciuojamas didziausias vidutinis km kiekis  float didz\_greitis = 0;  if (greitis1 > didz\_greitis) didz\_greitis = greitis1;  if (greitis2 > didz\_greitis) didz\_greitis = greitis2;  if (greitis3 > didz\_greitis) didz\_greitis = greitis3;  cout << "\nDidziausias vidutinis km skaicius: " << didz\_greitis;  //Funkcija baigiama  return 0;  } |
| --- |

### **Užduotis 7**

Sąlyga



Programa

| // --- Naudojamos bibliotekos ---  #include <iostream>  #include <fstream>  using namespace std;  // --- Pagrindine funkcija ---  int main(){  //Prieiga prie duomenu failo  ifstream input("Duomenys.txt");  if (!input){  cout << "Failas nerastas";  return 1;  }  //Sukuriami kintamieji  float komandos\_vidurkis; input >> komandos\_vidurkis;  int treniruotes; input >> treniruotes;  float vidurkis = 0; int temp;  //Pasiimami duomenys  for (int i = 0; i < treniruotes; i++){  input >> temp;  vidurkis += temp;  }  //Isvedami rezultatai  cout << "Zaidejas per treniruote imeta vidutiniskai " << vidurkis/treniruotes << " tasku.\n";  if (vidurkis/treniruotes > komandos\_vidurkis) cout << "Zaidejo vidurkis geresnis nei komandos";  //Baigiama funkcija  return 0;  } |
| --- |

Duomenu failas

| 16 (komandos tasku vidurkis)  4 (Treniruociu skaicius, toliau taskai)  10  18  7  12 |
| --- |

### 

### **Užduotis 4**

Sąlyga

Programa

|  |
| --- |

### **Užduotis 5**

Sąlyga

Programa

|  |
| --- |

### **Užduotis 6**

Sąlyga

Programa

|  |
| --- |

### 

### **Užduotis 7**

Sąlyga

Programa

|  |
| --- |

### **Užduotis 8**

Sąlyga

Programa

|  |
| --- |

### **Užduotis 9**

Sąlyga

Programa

|  |
| --- |