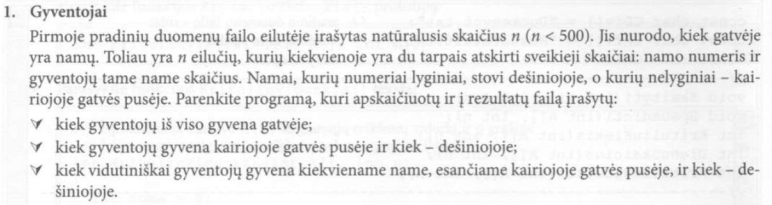
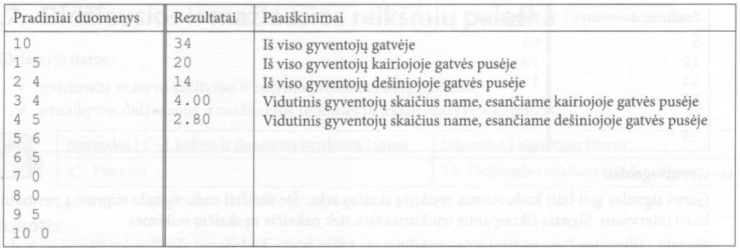
1.6 Pažintis su masyvu. Užduotys.

### **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.

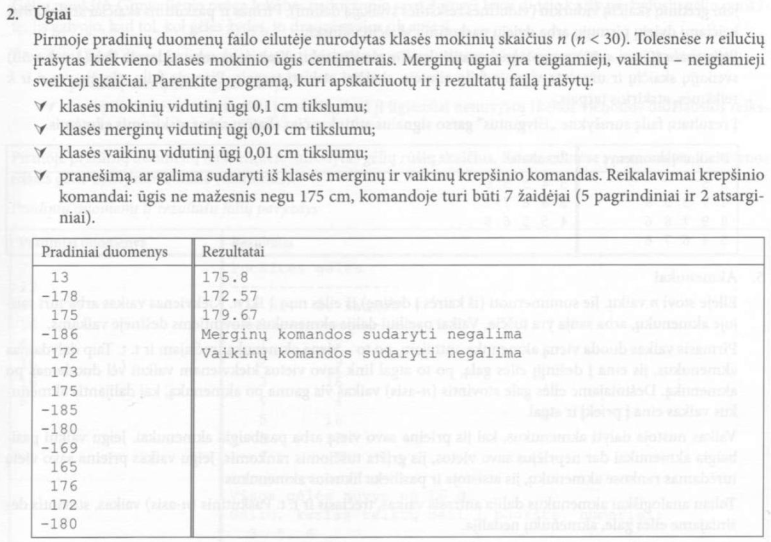
### **1.Gyventojai**





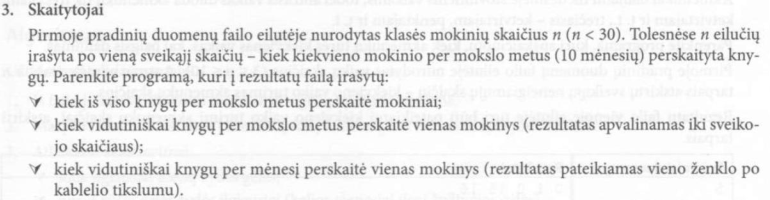
| #include <iostream> // Prijungiame biblioteką, kuri yra atsakinga už išvedimo bei įvedimo funkcijas  #include <fstream> // Prijungiame biblioteką, kuri yra atsakinga už failo sukūrimą, nuskaitymą bei išvedimą informacijos jame  using namespace std; // // trumpinama į std  int main() //pagrindine funkcija  {  ifstream input("input.txt");  ofstream output("output.txt"); //prisijungiam prie duomenu failo  if (!input)  {  cout << "Failas nerastas"; // IF funkcija, atsakinga už failo veikimą  return 0;  }  int namai; input >> namai; // kintamieji  float gyventojai[namai]; temp; kaires\_vidurkis; desines\_vidurkis  for (int i = 0; i < namai; i++) // sudedam į masyvą  {  input >> temp >> temp;  gyventojai[i] = temp;  }  int suma = 0;  for (int i = 0; i < namai; i++) suma += gyventojai[i];  output << suma << endl; // skaičiuojama gyventojų suma  suma = 0;  for (int i = 0; i < namai; i+=2) // skaičiuojame kairios pusės gyventojus ir vidurkį  {  suma += gyventojai[i];  temp++;  }  output << suma << endl;  kaires\_vidurkis = suma / temp; // skaiciuojame dešinės pusės gyventojus ir vidurkį  suma = 0; temp = 0;  for (int i = 1; i < namai; i+=2)  {  suma += gyventojai[i];  temp++;  }  output << suma << endl;  desines\_vidurkis = suma / temp;  output << kaires\_vidurkis << endl;  output << desines\_vidurkis << endl; // Vidurkių išvedimas  return 0;  } |
| --- |

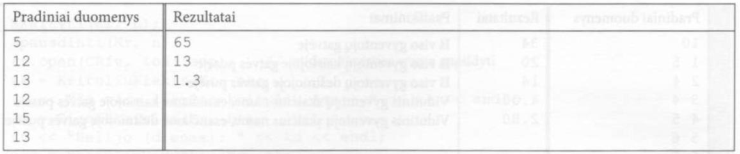
### **2.Ūgiai**



| #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  #include <cmath> // Prijungiame biblioteką, kuri yra atsakinga už matematinius veiksmus  #include <iomanip> // Prijungiame biblioteką, kuri yra atsakinga už input output  using namespace std; // trumpiname į std  int main() // pgr. funkcija  {  ifstream input("input.txt");  ofstream output("output.txt");    if (!input)  {  cout << "Failas nerastas"; // if funkcija, kuri patikrina ar failas veikia  return 0;  }    int zmones; input >> zmones;  int merg = 0; int vaik = 0; float vidurkism = 0; float vidurkisv = 0;  int mkomanda = 0; int vkomanda = 0; int temp; // kintamieji    for (int i = 0; i < zmones; i++)  {  input >> temp;  if (temp < 0)  {  temp = abs(temp);  vaik += temp;  vidurkisv++;  if (temp >= 175) vkomanda++;  }  else  {  merg += temp;  vidurkism++;  if (temp >= 175) mkomanda++; // tikrinama merginų ar vyrų komanda  }  }  output << fixed << setprecision(2) << (vaik + merg) / zmones << endl;  output << fixed << setprecision(2) << merg / vidurkism << endl;  output << fixed << setprecision(2) << vaik / vidurkisv << endl;  if (mkomanda >= 7) output << "Merginu komanda sudaryti galima" << endl;  else output << "Merginu komandos sudaryti negalima" << endl;  if (vkomanda >= 7) output << "Vaikinu komanda sudaryti galima";  else output << "Vaikinu komandos sudaryti negalima"; // Skaičiavimai ir išvedimas    return 0; // baigiam funkcija  } |
| --- |

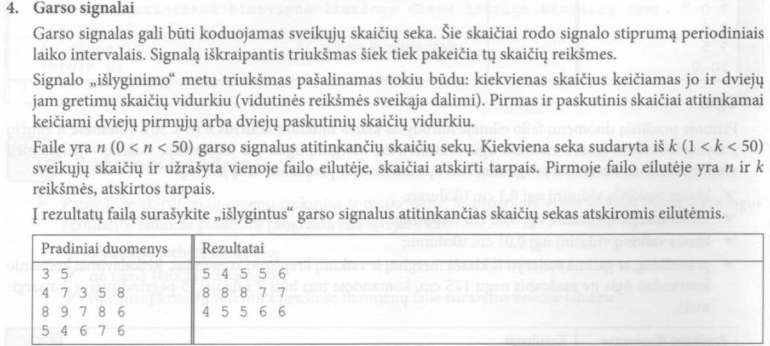
### **3.Skaitytojai**





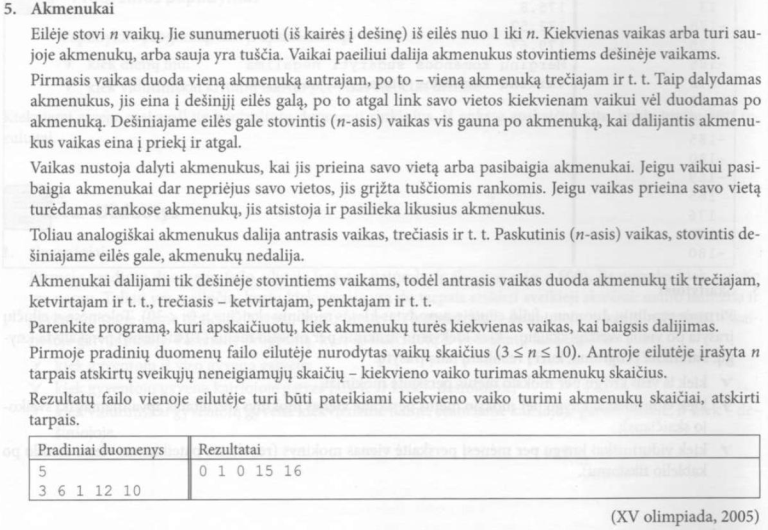
| #include <iostream> // Prijungiame biblioteką, kuri yra atsakinga už išvedimo bei įvedimo funkcijas  #include <fstream> // Prijungiame biblioteką, kuri yra atsakinga už failo sukūrimą, nuskaitymą bei išvedimą informacijos jame  #include <cmath> // Prijungiame biblioteką, kuri yra atsakinga už matematinius veiksmus  #include <iomanip> // Prijungiame biblioteką, kuri yra atsakinga už input output  using namespace std; // trumpiname į std  int main() // pgr. funkcija  {  ifstream input("input.txt");  ofstream output("output.txt");  if (!input)  {  cout << "Failas nerastas"; // tikrinam ar failas atsidaro  return 0;  }  int vaikai; int temp; float suma; // kintamieji sukuriami  input >> vaikai;    for (int i = 0; i < vaikai; i++)  {  input >> temp;  suma += temp; // knygų skaičius  }  output << suma << endl << suma / vaikai << endl;  output << suma / vaikai / 10; // Skaičiavimai ir išvedimas  return 0; // Funkcija baigta  } |
| --- |

### **4.Garso signalai**



|  |
| --- |

### **5.Akmenukai**



|  |
| --- |