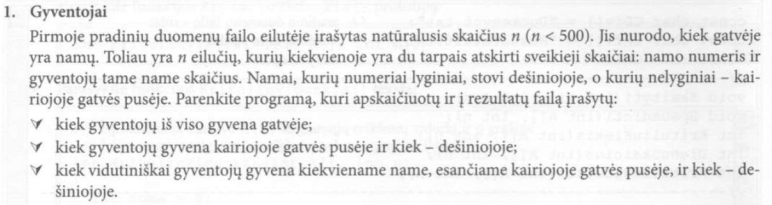
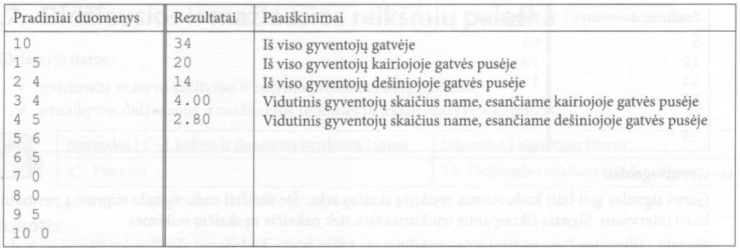
1.6 Pažintis su masyvu. Užduotys.

### **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.

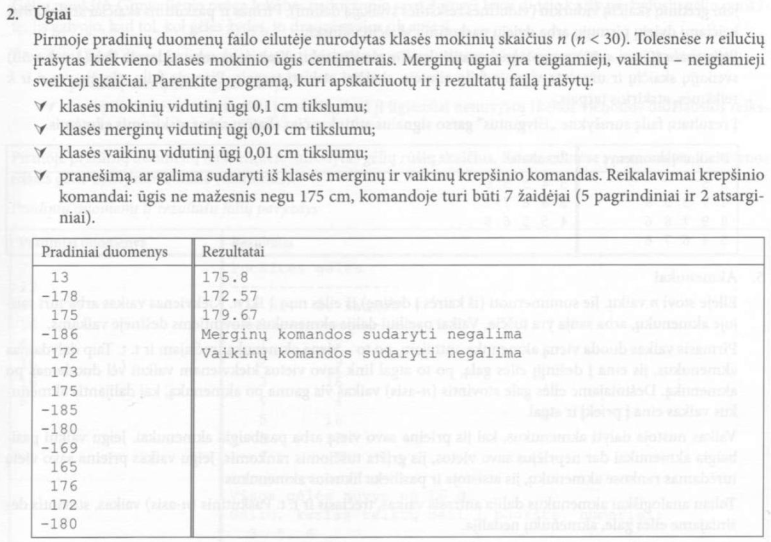
### **1.Gyventojai**





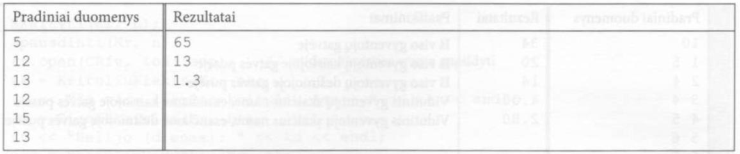
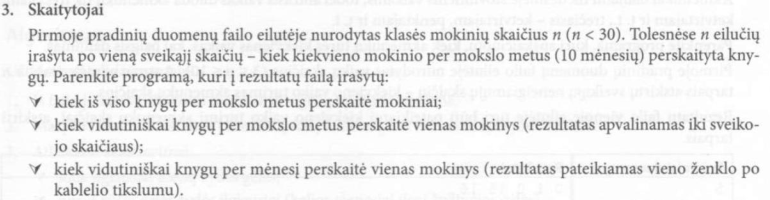
| //Prisidedame naudojamas bibliotekas  #include <iostream>  #include <fstream>  using namespace std;  //pagrindine funkcija  int main()  {  //prisijungiam prie duomenu failo  ifstream input("input.txt");  ofstream output("output.txt");  //patikrinam, ar duomenu failas egzistuoja  if (!input)  {  cout << "Failas nerastas";  return 0;  }  //sukuriami kintamieji  int namai; input >> namai;  float gyventojai[namai];  float temp;  float kaires\_vidurkis;  float desines\_vidurkis;  //kiekvieno namo gyventojus sudedame i masyva  for (int i = 0; i < namai; i++)  {  input >> temp >> temp;  gyventojai[i] = temp;  }  //skaiciuojame, kiek is viso yra gyventoju  int suma = 0;  for (int i = 0; i < namai; i++) suma += gyventojai[i];  output << suma << endl;  //skaiciuojame kairios puses gyventojus ir vidurki  suma = 0;  for (int i = 0; i < namai; i+=2)  {  suma += gyventojai[i];  temp++;  }  output << suma << endl;  kaires\_vidurkis = suma / temp;  //skaiciuojame desines puses gyventojus ir vidurki  suma = 0; temp = 0;  for (int i = 1; i < namai; i+=2)  {  suma += gyventojai[i];  temp++;  }  output << suma << endl;  desines\_vidurkis = suma / temp;  //isvedame vidurkius  output << kaires\_vidurkis << endl;  output << desines\_vidurkis << endl;  return 0;  } |
| --- |

### **2.Ūgiai**



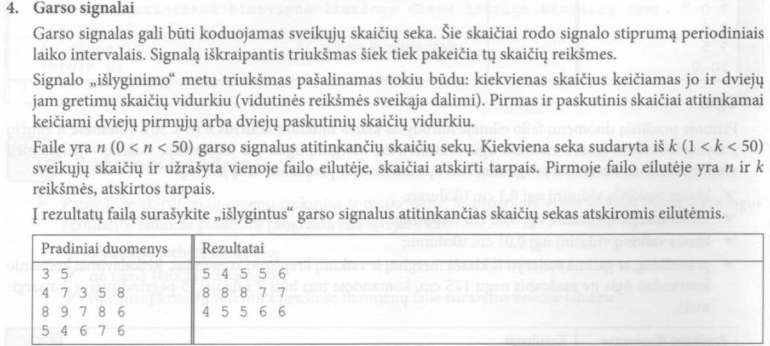
| //Prisidedame naudojamas bibliotekas  #include <iostream>  #include <fstream>  #include <cmath>  #include <iomanip>  using namespace std;  //pagrindine funkcija  int main()  {  //prisijungiam prie duomenu failo  ifstream input("input.txt");  ofstream output("output.txt");  //patikrinam, ar duomenu failas egzistuoja  if (!input)  {  cout << "Failas nerastas";  return 0;  }  //sukuriami kintamieji  int zmones; input >> zmones;  int merg = 0; int vaik = 0; float vidurkism = 0; float vidurkisv = 0;  int mkomanda = 0; int vkomanda = 0; int temp;  //surandama, ar tai vaikinas ar mergina  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++;  }  }  //duomenis suskaiciuojami, isvedami i rezultatu faila  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";  //funkcija baigiama  return 0;  } |
| --- |

### **3.Skaitytojai**



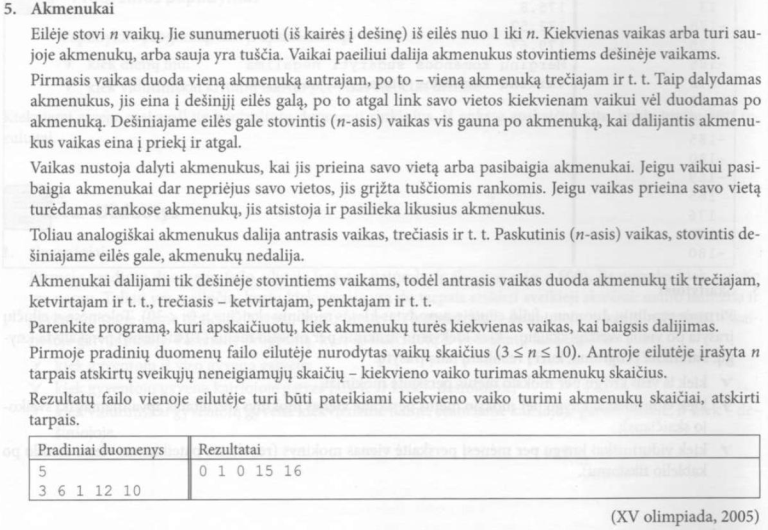
| //Prisidedame naudojamas bibliotekas  #include <iostream>  #include <fstream>  #include <cmath>  #include <iomanip>  using namespace std;  //pagrindine funkcija  int main()  {  //prisijungiam prie duomenu failo  ifstream input("input.txt");  ofstream output("output.txt");  //patikrinam, ar duomenu failas egzistuoja  if (!input)  {  cout << "Failas nerastas";  return 0;  }  //sukuriami kintamieji  int vaikai; int temp; float suma;  input >> vaikai;  //sudedamas knygu skaicius  for (int i = 0; i < vaikai; i++)  {  input >> temp;  suma += temp;  }  //suskaiciuojami ir isvedami duomenys  output << suma << endl << suma / vaikai << endl;  output << suma / vaikai / 10;  //funkcija baigiama  return 0;  } |
| --- |

### **4.Garso signalai**



|  |
| --- |

### **5.Akmenukai**



|  |
| --- |