1. **Užduotis.** **Sodinukai**

Pranas augina medelių sodinukus, kuriuos pardavinėja kitiems sodininkams. Pardavimui skirti sodinukai turi būti ne žemesni už **a1** cm ir ne aukštesni už **a2** cm. Prano medelyne auga **n** sodinukų. Rengdamasis rudeninei mugei, Pranas išmatavo kiekvieno sodinuko aukštį ir surašė duomenis faile **sodinukai\_data.txt**. Pirmoje eilutėje įrašytas sodinukų kiekis **n** (1 ≤ n ≤ 100). Sekančiose **n** eilučių įrašyti sodinukų aukščiai (sveikieji skaičiai). Paskutinėje eilutėje – du sveikieji skaičiai **a1** ir **a2**, nurodantys, kokiame aukščių intervale (imtinai) yra pardavimui tinkamų sodinukų aukščiai.

Parašykite programą, kuri rezultatų faile **sodinukai\_res.txt** išspausdintų:

* Vidutinį visų medelyne augančių sodinukų aukštį vieno skaitmens po kablelio tikslumu.
* Rudeninei mugei atrinktų sodinukų kiekį.
* Rudeninei mugei atrinktų sodinukų vidutinį aukštį vieno skaitmens po kablelio tikslumu.
* Rudeninei mugei atrinktų sodinukų aukščių sąrašą vienoje eilutėje (jeigu tokių sodinukų nėra, išspausdinti žodį 'NE' didžiosiomis raidėmis).

|  |  |
| --- | --- |
| sodinukai\_data.txt | sodinukai\_res.txt |
| 5  53  45  56  54  56  50 60 | 52.8  4  54.8  53 56 54 56 |

1. **Reikalavimai**

* Duomenų saugojimui naudoti statinius (!) masyvus.
* Aprašyti funkciją **void Skaityti()**, kuri nuskaito pradinius duomenis.
* Aprašyti funkciją **void Atrinkti()**, kuri sudeda rudeninei mugei atrinktus sodinukus į naują masyvą.
* Aprašyti funkciją **double Vidurkis()**, kuri grąžina vidutinį nurodytų sodinukų aukštį (į ją reikėtų kreiptis du kartus).
* Aprašyti funkciją **void Rasyti()**, kuri išspausdina rezultatus.

1. **Programos tekstas C++ kalba**

#include <iostream>

#include <iomanip>

#include <cmath>

#include <fstream>

using namespace std;

void Skaito (int A[], int &n, int &a, int &a1);

void Raso(int A[], int n, double vidvis, double vidpard, int k, int B[]);

void Atrinkti(int A[], int n, int a, int a1, int B[], int &k);

double Vidurkis(int A[], int n);

int main(){

int A[100], n, a, a1, B[100], k = 0;

Skaito(A, n, a, a1);

Atrinkti(A, n, a, a1, B, k);

Raso(A, n, Vidurkis(A, n), Vidurkis(B, k), k, B);

return 0;

}

void Skaito (int A[], int &n, int &a, int &a1)

{

ifstream df ("sodinukai\_data.txt");

df >> n;

for (int i = 0; i < n; i++)df >> A[i];

df >> a >> a1;

df.close();

}

void Raso(int A[], int n, double vidvis, double vidpard, int k, int B[])

{

ofstream rf ("sodinukai\_res.txt");

rf << fixed << setprecision(1) << vidvis << endl;

rf << k << endl;

rf << fixed << setprecision(1) << vidpard << endl;

if (B[0] == -1)rf << "NE" << endl;

else{

for(int i = 0; i < k; i++){

rf << B[i] << " ";

}

}

rf << endl;

rf.close();

}

void Atrinkti(int A[], int n, int a, int a1, int B[], int &k)

{

for (int i = 0; i < n; i++){

if(A[i] >= a && A[i] <= a1){

B[k] = A[i];

k++;

}

}

if (k == 0)B[0] = -1;

}

double Vidurkis(int A[], int n)

{

if (n != 0){

double s = 0;

for(int i = 0; i < n; i++)s = s + A[i];

return s / n;

}

else return 0;

}