1. **Užduotis.** **Paprasti uždaviniai**

Trijų ekspertų komisija programavimo konkursui nori parinkti keletą paprastų uždavinių, tačiau jie nesutaria, kurie už pasiūlytų uždavinių yra paprasti. Kad pasiektų sutarimą, ekspertai nusprendžia naudoti tokią vertinimo sistemą: Kiekvienas ekspertas kiekvieną iš siūlomų uždavinių įvertina sudėtingumo balu 0 ≤ b ≤ 100 (keletas uždavinių gali būti vertinami vienodu balu). Galiausiai palyginami visų ekspertų vertinimai – paprastu laikomas toks uždavinys, kurį mažiausiu balu įvertino bent 2 ekspertai iš 3, ir nei vienas neįvertino didžiausiu balu. Pagal ekspertų vertinimus nustatykite, kurie uždaviniai yra paprasti.

Tekstinio failo **paprasti\_uzdaviniai\_data.txt** pirmoje eilutėje nurodytas siūlomų uždavinių skaičius **n**. Tolesnėse trijose eilutėse surašyti kiekvienam uždaviniui ekspertų skirti sudėtingumo balai – vienam ekspertui viena eilutė.

Rezultatus įrašykite į failą **paprasti\_uzdaviniai\_res.txt**, kuriame išvardinkite paprastų uždavinių numerius (uždaviniai numeruojami nuo 1).

Paaiškinimas: 1-ąjį uždavinį mažiausiais balais įvertino pirmasis (1) ir antrasis (5) ekspertai. 3-ąjį uždavinį mažiausiais balais įvertino pirmasis (1) ir trečiasis (8) ekspertai. 5-asis uždavinys nėra paprastas, nes jį didžiausiu balu (13) įvertino antrasis ekspertas.

|  |  |
| --- | --- |
| paprasti\_uzdaviniai\_data.txt | paprasti\_uzdaviniai\_res.txt |
| 5  1 2 1 7 1  5 9 6 8 13  9 10 8 15 8 | 1 3 |

1. **Reikalavimai**

* Naudokite vienmačius sveikųjų skaičių masyvus.
* Pradinių duomenų skaitymo funkcija **void**.
* Didžiausios reikšmės paieškos funkcija, grąžinanti reikšmę per funkcijos vardą.
* Mažiausios reikšmės paieškos funkcija, grąžinanti reikšmę per funkcijos vardą.
* Rezultatų rašymo į failą funkcija **void**.

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

#include <iostream>

#include <iomanip>

#include <cmath>

#include <fstream>

void Skaito(int P[], int A[], int T[], int &n);

int Maziausias(int A[], int n);

int Didziausias(int A[], int n);

void Paprasti(int P[], int A[], int T[], int n, int U[], int &ind);

void Raso(int A[], int n);

using namespace std;

int main(){

int P[100], A[100], T[100], n, U[100], ind = 0;

Skaito(P, A, T, n);

Paprasti(P, A, T, n, U, ind);

Raso(U, ind);

return 0;

}

void Skaito(int P[], int A[], int T[], int &n)

{

ifstream df ("paprasti\_uzdaviniai\_data.txt");

df >> n;

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

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

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

df.close();

}

int Maziausias(int A[], int n)

{

int ind = 0;

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

return A[ind];

}

int Didziausias(int A[], int n)

{

int ind = 0;

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

return A[ind];

}

void Paprasti(int P[], int A[], int T[], int n, int U[], int &ind)

{

int pm = Maziausias(P, n), am = Maziausias(A, n), tm = Maziausias(T, n), pd = Didziausias(P, n), ad = Didziausias(A, n), td = Didziausias(T, n);

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

if(P[i] == pm && A[i] == am && T[i] != td || P[i] == pm && T[i] == tm && A[i] != ad || T[i] == tm && A[i] == am && P[i] != pd){

U[ind] = i + 1;

ind++;

}

}

}

void Raso(int A[], int n){

ofstream rf ("paprasti\_uzdaviniai\_res.txt");

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

rf << A[i] << " ";

}

rf.close();

}