1. **Užduotis.** **Protmūšis**

Protmūšyje dalyvauja **n** komandų. Kiekviena komanda turėjo atsakyti į **m** klausimų. Pradinių duomenų failo **protmusis\_data.txt** pirmoje failo eilutėje įrašytas komandų skaičius n ir klausimų skaičius m, likusiose n eilučių įrašytas komandos pavadinimas (jį sudaro vienas žodis) ir m sveikųjų skaičių: 0 – jei į klausimą komanda neatsakė, 1 – jei komanda į klausimą atsakė teisingai.

Parašykite programą, kuri į rezultatų failą **protmusis\_res.txt** išvestų rezultatus, kaip pateikta pavyzdyje: nurodomas komandos pavadinimas, bei į kelis klausimus komandos atsakė teisingai ir į kelis klaidingai. Toliau faile surašomas komandų, **atsakiusių į daugiau kaip pusę klausimų teisingai**, sąrašas.

Jei nėra tokių komandų, tuomet faile turi būti įrašytas žodis „nera“.

|  |  |
| --- | --- |
| protmusis\_data.txt | protmusis\_res.txt |
| 5 6  Zalieji 1 1 1 0 1 1  Melynieji 0 0 1 1 1 1  Raudonieji 1 1 1 1 0 1  Geltonieji 0 0 0 1 1 1  Baltieji 1 1 1 1 1 1 | VISOS KOMANDOS:  Zalieji 5 1  Melynieji 4 2  Raudonieji 5 1  Geltonieji 3 3  Baltieji 6 0  GERIAUSIOS KOMANDOS:  Zalieji 5 1  Melynieji 4 2  Raudonieji 5 1  Baltieji 6 0 |

1. **Reikalavimai**

* Naudojamas masyvo duomenų tipas.
* Pradinių duomenų skaitymo funkcija **void Skaitymas()**.
* Vienos komandos teisingai ir klaidingai atsakytų klausimų skaičiaus skaičiavimo funkcija **void Atsakymai()**, grąžinanti apskaičiuotas reikšmes per parametrus.
* Komandų, atsakiusių į daugiau kaip pusę klausimų teisingai, atrinkimo funkcija **void Geriausios()**. Funkcija turi suformuoti geriausių komandų masyvą.
* Rezultatų rašymo funkcija **void Rasymas()**.

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

#include <iostream>

#include <iomanip>

#include <cmath>

#include <fstream>

using namespace std;

void Skaito(string A[], int B[], int &n, int &m);

void Atsakymai(int B[], int &m, int k, int &t, int &n);

void Geriausios(int T[], int NT[], int &n, int G[], int &ind);

void Raso(string A[], int T[], int NT[], int &n, int G[], int &ind);

int main(){

string A[100];

int B[100], n, m, t = 0, nt = 0, T[100], N[100], G[100], ind = 0;

Skaito(A, B, n, m);

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

Atsakymai(B, m, i, t, nt);

T[i] = t;

N[i] = nt;

t = 0;

nt = 0;

}

Geriausios(T, N, n, G, ind);

Raso(A, T, N, n, G, ind);

return 0;

}

void Skaito(string A[], int B[], int &n, int &m)

{

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

df >> n >> m;

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

df >> A[i];

for(int i1 = 0; i1 < m; i1++)df >> B[i1 + i \* m];

}

df.close();

}

void Atsakymai(int B[], int &m, int k, int &t, int &n)

{

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

if(B[i + k \* m])t++;

else n++;

}

}

void Geriausios(int T[], int NT[], int &n, int G[], int &ind){

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

if(T[i] > NT[i]){

G[ind] = i;

ind++;

}

}

}

void Raso(string A[], int T[], int NT[], int &n, int G[], int &ind)

{

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

rf << "VISOS KOMANDOS:" << endl;

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

rf << A[i] << " " << T[i] << " " << NT[i] << endl;

}

rf << "GERIAUSIOS KOMANDOS:" << endl;

if (ind > 0){

for (int i = 0; i < ind; i++)rf << A[G[i]] << " " << T[G[i]] << " " << NT[G[i]] << endl;

}

else rf << "nera";

rf.close();

}