1. **Užduotis.** **Kelionė**

Tomas keliaudamas po Lietuvą, kelionės gale norėjo išsiaiškinti, koks visą kelionę buvo jo vidutinis automobilio greitis.

Parenkite programą, kuri padėtų Tomui išsiaiškinti, koks buvo jo vidutinis greitis.

**Pradiniai duomenys:**

Skaičius **n**, rodantis kiek skirtingų tarpų buvo, kai jo greitis kito. Kitose **n** eilučių yra **2** skaičiai - pirmas, rodantis atstumą (**metrais**), kurį nuvažiavo Tomas, o antras - laiko tarpas (**sekundėmis**) per kurį nuvažiavo tą atstumą. Skaičiai atskirti 1 tarpo simboliu.

Apskaičiuokite, koks buvo Tomas vidutinis greitis kelionės metu.

Rezultatų faile išspausdinkite vidutinį greitį m/s ir km/h atskirtus tarpo simboliu (**2 skaičių po kablelio tikslumu**).

|  |  |
| --- | --- |
| kelione.txt | rezultatai.txt |
| 5  3000 120  1500 70  2200 180  500 20  1300 45 | 19.54 70.34 |

1. **Reikalavimai**

* 2 masyvai atstumui laikyti bei laikui.
* **void Skaityti()** funkcija, duomenims skaityti.
* **int Suma()** funkcija, sumuojanti visą atstumą bei visą laiką.
* **double Vidurkis()** funkcija, skaičiuojanti vidutinį greitį.
* **void Spausdinti()** funkcija, rezultatams į failą spausdint.

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

#include <iostream>

#include <iomanip>

#include <cmath>

#include <fstream>

using namespace std;

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

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

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

void Raso(double vidm, double vidkm);

int main(){

int n, K[100], L[100];

Skaito(n, K, L);

Raso(Vidurkis(n, K, L), Vidurkis(n, K, L) \* 3600 / 1000);

return 0;

}

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

{

ifstream df ("kelione.txt");

df >> n;

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

df.close();

}

int Suma(int n, int A[])

{

int s = 0;

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

return s;

}

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

{

double vid;

vid = Suma(n, A);

return vid / Suma(n, B);

}

void Raso(double vidm, double vidkm)

{

ofstream rf ("rezultatai.txt");

rf << fixed << setprecision(2) << vidm << " " << vidkm;

rf.close();

}