1. **Užduotis.** **Dalelė**

Mokslininkas stebi vandens paviršiuje pakibusios kietosios dalelės judėjimą ir kas **N** sekundžių užrašo jos koordinates. Parašykite programą, kuri apskaičiuotų kietosios dalelės mažiausią, vidutinį, didžiausią greitį ir dalelės nukeliautą kelią.

Tekstiniame faile **dalele\_data.txt** pirmoje eilutėje užrašytas dalelės koordinačių skaičius **M** (**1 < M ≤ 100**) ir koordinačių fiksavimo intervalas **N** (**N > 0**) sekundėmis. Sekančiose failo eilutėse užrašytos dalelės koordinatės **Xi** ir **Yi** (**1 ≤ i ≤ M**) milimetrais (**Xi** ir **Yi** – realieji skaičiai).

Tekstiniame faile **dalele\_res.txt** spausdinkite kietosios dalelės mažiausią, vidutinį, didžiausią greitį ir dalelės nukeliautą kelią kaip nurodyta pavyzdyje.

|  |  |
| --- | --- |
| dalele\_data.txt | dalele\_res.txt |
| 5 2  1.8 2.7  -0.5 4.3  -1.4 3.2  0.1 1.4  0.7 -0.5 | Maziausias greitis: 0.71 mm/s  Vidutinis greitis: 1.07 mm/s  Didziausias greitis: 1.40 mm/s  Nukeliautas kelias: 8.56 mm |

1. **Reikalavimai**

* Naudokite realiųjų skaičių masyvus.
* Pradinių duomenų skaitymo funkcija **void**.
* Funkcija, surandanti mažiausią greitį.
* Funkcija, skaičiuojanti vidutinį greitį.
* Funkcija, surandanti didžiausią greitį.
* Funkcija, skaičiuojanti nukeliautą kelią.

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

#include <iostream>

#include <iomanip>

#include <cmath>

#include <fstream>

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

void Greitis(double A[], double B[], int n, int l, double D[]);

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

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

double Kelias(double A[], int n, int l);

double Vidutinis(double A[], int n, int l);

using namespace std;

int main(){

double A[200], B[200], D[200];

int n, l;

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

Skaito(A, B, n, l);

Greitis(A, B, n, l, D);

rf << "Maziausias greitis: " << fixed << setprecision(2) << Maziausias(D, n - 1) << " mm/s" << endl << "Vidutinis greitis: " << Vidutinis(D, n - 1, l) << " mm/s" << endl;

rf << "Didziausias greitis: " << fixed << setprecision(2) << Didziausias(D, n - 1) << " mm/s" << endl << "Nukeliautas kelias: " << Kelias(D, n - 1, l) << " mm" << endl;

rf.close();

return 0;

}

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

{

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

df >> n >> l;

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

df.close();

}

void Greitis(double A[], double B[], int n, int l, double D[])

{

for (int i = 0; i < n - 1; i++)D[i] = sqrt(pow(A[i] - A[i + 1], 2) + pow(B[i] - B[i + 1], 2)) / l;

}

double Maziausias(double A[], int n)

{

int ind = 0;

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

return A[ind];

}

double Didziausias(double A[], int n)

{

int ind = 0;

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

return A[ind];

}

double Kelias(double A[], int n, int l)

{

double s = 0;

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

return s;

}

double Vidutinis(double A[], int n, int l)

{

return Kelias(A, n, l)/ (l \* n);

}