МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Звіт з лабораторної роботи №5

з предмету «Алгоритмізація та програмування»

Виконав:

Студент групи КН-36А

Кулик В.В

Перевірив:

доц. каф. ПІ ІТУ

Захарова Т.В.

Харків 2016

Лабораторна робота №5

**Тема**: Робота зі структурами та файлами.

### Завдання:

### 1.1 Точки у тривимірному просторі

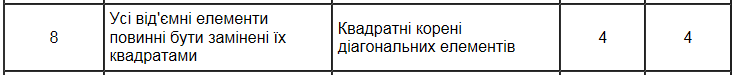
Написати програму, яка обчислює відстань між двома точками в тривимірному просторі.

### 1.2 Середнє арифметичне

Написати програму, яка зчитує значення з плаваючою точкою з текстового файлу до кінця файлу і обчислює середнє арифметичне цих значень.

### 1.3 Індивідуальне завдання

Написати програму, яка забезпечує файлове введення та виведення і включає індивідуальне завдання [попередньої лабораторної роботи](http://iwanoff.96.lt/algorithmization/LabTraining04.htm). Слід реалізувати такі дії:

* визначення константи (n) яка визначає кількість стовпців двовимірного масиву
* відкриття файлу для читання (файл повинен бути підготовлений за допомогою текстового редактора)
* читання цілих чисел до кінця файлу і зберігання їх у зв'язаному списку
* створення двовимірного масиву в динамічній пам'яті; кількість рядків повинна бути обчислена на основі кількості зчитаних з файлу значень та визначеної стовпців
* заповнення двовимірного масиву рядок за рядком; відсутні елементи останнього рядка повинні бути заповнені нулями
* видалення елементів зв'язаного списку з динамічної пам'яті
* реалізація [попереднього індивідуального завдання](http://iwanoff.96.lt/algorithmization/LabTraining04.htm)
* зберігання результатів в новому файлі
* видалення масивів операцією **delete**.

**Основна частина:**

**Завдання №1**

#include <iostream>

#include <cmath>

#include <stdio.h>

using namespace std;

struct Point

{

int x, y, z;

};

double sqr(double x)

{

return x \* x;

}

double distance(Point p1, Point p2)

{

return sqrt(sqr(p1.x - p2.x) + sqr(p1.y - p2.y) + sqr(p1.z - p2.z));

}

int main()

{

Point p1 = { 2, 4, 6 };

Point p2 = { 1, 3, 5 };

printf("%f\n", distance(p1, p2));

system("pause");

return 0;

}

**Завдання №2**

#include <iostream>

#include <fstream>

#include <stdio.h>

using namespace std;

int main()

{

ifstream in("Numbers.txt");

double x, i = 0, sum = 0;

while (in >> x)

{

sum += x;

i += 1;

}

printf("%f\n",sum/i);

system("pause");

return 0;

}

**Завдання №3**

#include<iostream>

#include<fstream>

using namespace std;

struct li

{

int data;

li \*next;

};

int min(int n, int b);

void outRes(ofstream& obj, double \*B, int m);

li\* deleteList(li \*obj, li \*head);

void indTask(int m, int n, li \*obj, li \*head, double \*B, ofstream& output);

li\* iniList(int m, int n, int \*\*A, li\* obj, li\* head, ifstream& input);

void delArr(int m, int \*\*A, double \*B); int \*\*makeNewMem(int \*\*A, int m, int n);

void closeStream(ifstream& in, ofstream& out); void getM(ifstream& in, int& m, int& n);

double \*makeNewMem(double \*B, int n);

li\* reverse(li\*head, int m, int n);

int main() {

li\* head = 0;

li\* p = 0;

int m = 0, n = 0;

ifstream in("read.txt");

ofstream out;

out.open("res.txt");

getM(in, m, n);//Получаем кол-во рядков

int \*\*A = 0;

A = makeNewMem(A, m, n);//Выделяем память для двохмерного массива

double \*B = 0;

B = makeNewMem(B, m);//Выделяем память для одномерного массива

head = iniList(m, n, A, p, head, in);//Получаем адрес головы списка

indTask(m, n, p, head, B, out);//Делаем индивидуальное задание

head = deleteList(p, head);//Удаляем список

outRes(out, B, m);//Выводим результат

delArr(m, A, B);//Удаляем 2 массива

closeStream(in, out);//Закрываем потоки

return 0;

}

void outRes(ofstream& obj, double \*B, int m)

{

for (int l = 0; l <m; l++)

{

obj << l + 1 << " sqrt("<<pow(B[l], 2)<<"): " << B[l] << endl;

}

}

li\* deleteList(li \*obj, li \*head)

{

obj = head;

while (head)

{

obj = head;

head = obj->next;

delete obj;

}

delete head;

return 0;

}

void indTask(int m, int n, li \*obj, li \*head, double \*B, ofstream& output)

{

double sq;

obj = head;

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

{

for (int j = 0; j < n; j++)

{

if (obj->data < 0)

{

obj->data \*= obj->data;

}

if (i == j)

{

sq = sqrt(obj->data);

B[i] = sq;

}

obj = obj->next;

}

}

}

li \* iniList(int m, int n, int \*\*A, li\*obj, li\*head, ifstream& input)

{

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

{

for (int j = 0; j < n; j++)

{

input >> A[i][j];

if (!input) { A[i][j] = 0; }

}

}

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

{

for (int j = 0; j < n; j++)

{

obj = new li;

obj->data = A[i][j];

obj->next = head;

head = obj;

}

}

return reverse(head, m, n);

}

li\* reverse(li\*head, int m, int n)

{

li\*obj;

li\*tempHead = head;

li\*temp2 = head;

for (int i = 0; i < m\*n; i++)

{

obj = new li;

obj->data = temp2->data;

obj->next = tempHead;

tempHead = obj;

temp2 = temp2->next;

}

//head = tempHead;

return tempHead;

}

void delArr(int m, int \*\*A, double \*B)

{

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

{

delete[] A[i];

}

delete[]A;

//delete[]B;

}

int \*\*makeNewMem(int \*\*A, int m, int n)

{

A = new int\*[m];

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

{

A[i] = new int[n];

}

return A;

}

void closeStream(ifstream& in, ofstream& out)

{

in.close();

out.close();

}

void getM(ifstream& in, int& m, int& n)

{

int x = 0;

int y = 1;

in >> n;

while (in >> x)

{

if (y % (n + 1) == 0 || y == 1)

{

m++; y = 1;

}

y++;

}

in.close();

in.open("read.txt");

in >> n;

}

double \*makeNewMem(double \*B, int m)

{

B = new double[m];

return B;

}

return 0;

}

**Висновок: Навчився працювати зі структурами та файлами. Написав програми, які зчитують дані з файлів, створюють зв’язний список та працюють з його елементами. Дізнався чим зв’язний список краще масиву, та які він має недоліки.**

**1.**

#include <iostream>

struct Point

{

int x, y;

};

double distance(Point p)

{

return p.x \* p.y;

}

int main()

{

Point p = { 4, 6 };

std::cout << distance(p) << "\n";

system("pause");

return 0;

}

**2.**

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ifstream in("Numbers.txt"); // 1 2 3 4 5 6 0 7 8 9

int x, dob = 1;

while (in >> x)

{

if (x != 0)

{

dob \*= x;

}

else

{

break;

}

}

cout << dob << "\n";

system("pause");

return 0;

}

**3.**

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ifstream in("Numbers.txt"); //1 2 3 4 5 6 0 7 8 9

int x, dob = 1;

while (in >> x)

{

if (x != 0)

{

dob \*= x;

}

}

cout << dob << "\n";

system("pause");

return 0;

}

**4.**

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

const int n = 10;

int a[n] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

int sum = 0;

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

{

if ((a[i] % 2) != 0)

{

sum += a[i];

}

}

ofstream out("Numbers.txt");

out << sum;

system("pause");

return 0;

}