**Задание 1.**

Написать программу, которая считывает текст из файла и выводит на экран только строки, содержащие числа (могут быть вещественными). Считывать текста с помощью классов для работы с файловыми потоками, использовать возможности класса string. Форматирование вывода строк-чисел: если число целое, то форматировать по левому краю, ширина поля 7, заполняющие символы - ‘#’; если число вещественное, то вывод в виде мантиссы с порядком, точность 3, ширина поля 14, заполняющие символы - ‘\_’.

Реализация в коде:

#include <stdio.h>

#include <iostream>

#include <fstream>

#include <iomanip>

#include <stdlib.h>

using namespace std;

void StrToInt(string s) //функция, которая принимает элемент из файла, проверяет на число-образность и выводит если число

{

int temp=0;

int i = 0;

int sign = 0;

double d;

if (s[i] == '-') //проверка на отрицательное число

{

sign = 1;

i++;

}

while (s[i] >= 0x30 && s[i] <= 0x39 || s[i]=='.') //проверка на число

{

if(s[i]=='.') { //вещественное число или нет

temp=1;

}

i++;

}

if(s[i]=='\0') { //если это число, то вывод

if(temp==1){ //вывод вещественного числа

istringstream(s) >> d; //это важно

cout.setf(ios::scientific);

cout.fill('\_');

cout.precision(3);

cout.width(14);

cout<<d<<endl;

} else { //вывод целого числа

cout.setf(ios::left);

cout.fill('#');

cout.width(7);

cout<<s<<endl;

cout.unsetf(ios::left); //без этого будет некорректный вывод

}

}

}

int main()

{

setlocale (LC\_ALL, "RUS"); //СО-Ю-Ю-Ю-ЮЗ НЕ РУШИ-И-ИМЫЙ...

int n=0;

string a;

fstream F;

F.open("baz.txt"); //открыл файл

if (F) //ЕСЛИ ОТКРЫЛСЯ

{

while (!F.eof()) //идем до конца файла по циклу

{

F>>a; //перенос элемента в строку

StrToInt(a); //вызов функции

}

F.close(); //закрытие файла

}

else cout<<"The file does not exist"<<endl;

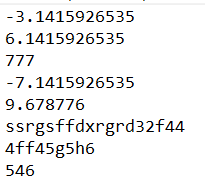
system("pause");

return 0;

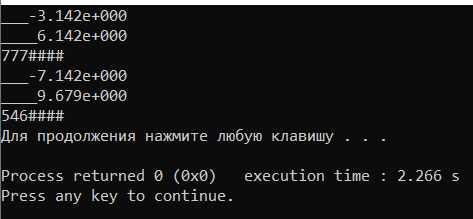
}

Проверка работы:

Данный в файле:



Результат выполнения программы:



**Задание 2**

Написать программу, которая сначала считывает числа из файла в первый вектор, затем сортирует их по возрастанию, последовательно сохраняя в другой вектор и удаляя первый. Вывод отсортированного вектора на экран произвести с помощью итератора.

Реализация в коде:

#include <stdio.h>

#include <iostream>

#include <fstream>

#include <iomanip>

#include <stdlib.h>

#include <vector>

#include <list>

#include <deque>

using namespace std;

int main()

{

vector<int>vec; //первый вектор

vector<int>vec1; //второй вектор

ifstream F("input.txt"); //файл для чтения(по умолчанию)

int ch;

int n;

while (!F.eof()) //цикл, чтобы загрузить элементы файла в вектор 1

{

F >> ch;

vec.push\_back(ch);

}

F.close(); //закрытие файла

int temp=vec[0]; //это важно

int m=vec.size(); //это тоже

for(int i=0;i<m-1;i++) { //цикл, что ищет мнимальный элемент вектора, потом загружает его во второй вектор и удаляет из первого(кроме последнего элемента)

for(int j=0;j<vec.size();j++) {

if (vec[j]<temp) {

temp=vec[j];

n=j;

}

}

vec1.push\_back(temp); //пихаем минимальный элемент в вектор

vec.erase(vec.begin()+n); //удаляем его из первого вектора

temp=vec[0];

}

vec1.push\_back(temp); //тоже самое с последним элементом

vec.clear();

for (vector<int>::iterator i = vec1.begin(); //радуемся, что работает скопированный из лекции вывод итератором

i != vec1.end(); ++i)

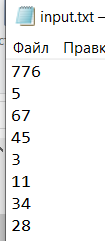
cout << \*i << endl;

return 0;

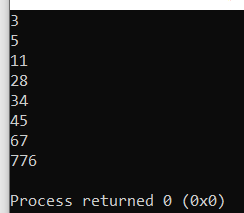
}

Проверка работы:

Данный в файле:



Результат выполнения программы:



Все получилось, ура)