**Министерство образования и науки Российской Федерации**

**Пермское федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский Национальный Исследовательский Политехнический Университет»**

**Электротехнический факультет**

**Кафедра «Информационные технологии и автоматизированные системы»**

**ОТЧЁТ**

По лабораторной работе №9 на тему

«Строковый ввод-вывод»

Вариант №11

Выполнил студент группы ИВТ-20-2б

Сабуров Павел Алексеевич

Проверил доцент кафедры ИТАС

Полякова О.А.

Пермь 2021

**Цель работы** –научиться работать с текстовыми файлами, ввод-вывод текстовой информации и её хранение на внешних носителях.

**Постановка задачи**

Задача – выполнить следующие действия:

1. Создать текстовый файл F1 не менее, чем из 10 строк;
2. Записать в него информацию;
3. Реализовать программу, которая выполняет задачи;

Исходные данные для **варианта №11**:

Задачи:

1. Скопировать из файла F1 в файл F2 все строки, кроме той строки, которая содержит самое короткое слово;
2. Напечатать в консоль номер этой строки;

**Анализ задачи**

Для того, чтобы решить задачу, нужно написать программу, которая выполняет следующие действия:

1. Открывает файл для чтения;
2. Проходит по каждой строке файла
3. Находит длину самого короткого слова в данной строке (с учётом знаков препинания);
4. Сравнивает длины строк и находит строку с самым коротким словом;
5. Сохраняет индекс строки с самым коротким словом;
6. Возвращается на начало файла;
7. Открывает второй файл для записи;
8. Проходит по каждой строке файла;
9. Копирует строку во второй файл при условии, что индекс строки не равен индексу строки с самым коротким словом;
10. Печатает на консоль индекс строки с самым коротким словом;

Для решения задачи были использованы следующие средства:

1. Язык программирования C++ (Microsoft Visual C++)
2. Текстовый редактор Microsoft Visual Studio Code

В первую очередь, программа будет работать с файлами, строками и целочисленными счётчиками (с помощью которых будет происходить расчёт длин слов в строке), а именно с такими типами данных, как:

* int (целое число)
* string (строка)
* char (символ)
* ifstream (класс, в котором происходит чтение из файла)
* ofstream (класс, в котором происходит чтение в файл)

Для того, чтобы успешно использовать классы ifstream и ofstream, необходимо подключить заголовочный файл fstream. Для ввода-вывода в консоль, нужно подключить файл iostream, а для работы со строками, string.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

Чтобы программа успешно открыла файл для чтения и записывала данные в файл, объявляются классы ifstream и ofstream. В качестве аргумента, принимается путь к файлам, с которыми будет происходить работа. Для удобства, рабочие файла были выведены в отдельную папку Work files.

ifstream inputFile("Work files\\F1.txt");

ofstream outputFile("Work files\\F2.txt");

Поскольку в задаче требуется скопировать все строки, кроме той, которая содержит самое короткое слово, и напечатать номер этой строки, необходимо объявить переменные, в которых будет сохранены:

* Номер (индекс) строки с самым коротким словом
* Длина самого короткого слова

Также объявляется переменная с рабочим индексом для того, чтобы обратиться к каждой строке (изначально количество строк не известно).

int shortestWordIndex,

shortestWordLength = INT\_MAX,

workIndex = 0;

Далее идёт проверка на то, успешно ли файл был открыт (если файл открыт успешно, то начинается чтение из файла, в противном случае выводится сообщение об ошибке, копирование строки в рабочую строку, расчёт длины самого короткого слова в строке и сравнение с длиной самого короткого найденного слова (изначально значение самого короткого слова равна максимальному возможному числу).

В цикле while идёт проверка на то, достигла ли программа конца файла: eof = true, если конец файла достигнут.

Функция getline отвечает за чтение строки (в том числе принимаются пробелы).

Дополнительно идёт вывод полученных строк и длин самых коротких слов.

if (inputFile.is\_open())

{

while (!inputFile.eof())

{

workIndex++;

string row;

getline(inputFile, row);

int minLength = ShortWordLength(row);

if (minLength < shortestWordLength)

{

shortestWordIndex = workIndex;

shortestWordLength = minLength;

}

cout << "row #" << workIndex << ": \"" << row << "\": "

<< minLength << endl;

}

//the next code will be here

}

else

{

cout << "Error: the file is not open!!!" << endl << endl

<< "Please, check does Work files\\F1.txt exist"

<< endl;

}

Для того, чтобы находить длину самого короткого слова в строке, была реализована функция ShortWordLength, которая в качестве аргумента принимает строку, которую нужно проанализировать. Поскольку знаки препинания, пробелы и другие спецсимволы не являются частью слов, необходимо проверить, является ли текущий символ символом слова. За это отвечает реализованная функция IsWordLetter, которая возвращает значение типа bool (true – является буквой, false – не является буквой).

bool IsWordLetter(char x)

{

return (toupper(x) != tolower(x));

}

int ShortWordLength(string expression)

{

int

wordLength = 0,

shortWordLength = INT\_MAX;

for (int i = 0; i < expression.length(); i++)

{

if (IsWordLetter(expression[i]))

wordLength++;

else

{

if ((wordLength < shortWordLength) && (wordLength != 0))

shortWordLength = wordLength;

wordLength = 0;

}

}

return shortWordLength;

}

После того, как программа полностью прошла по файлу, необходимо вернуться в начало файла и обнулить переменную, отвечающую за рабочий индекс строки (номер строки, с которой происходит работа на текущей итерации). За возврат в начальное положение, отвечает функция seekq, которая отвечает за то, в какую позицию перенести чтение файла (в начало, в n-й байт или в конец файла)

workIndex = 0;

inputFile.seekg(ios\_base::beg);

Теперь программа снова проходит через каждую строку, но сравнивает текущей индекс с индексом строки с самым коротким словом (если индекс слова равен таковому, то он не выводится в файл F2). Вывод в файл F2 – построчный, как и чтение из F1.

while (!inputFile.eof())

{

workIndex++;

string row;

getline(inputFile, row);

if (workIndex != shortestWordIndex)

outputFile << row << endl;

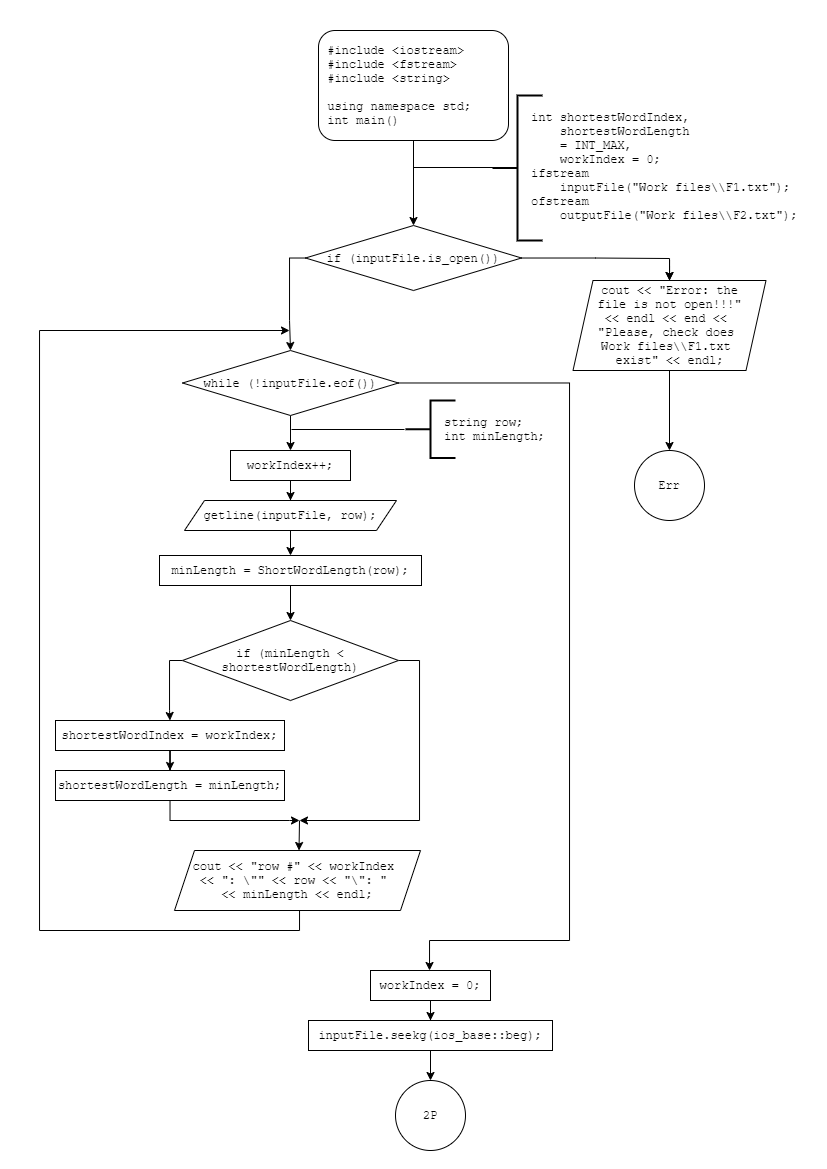
}

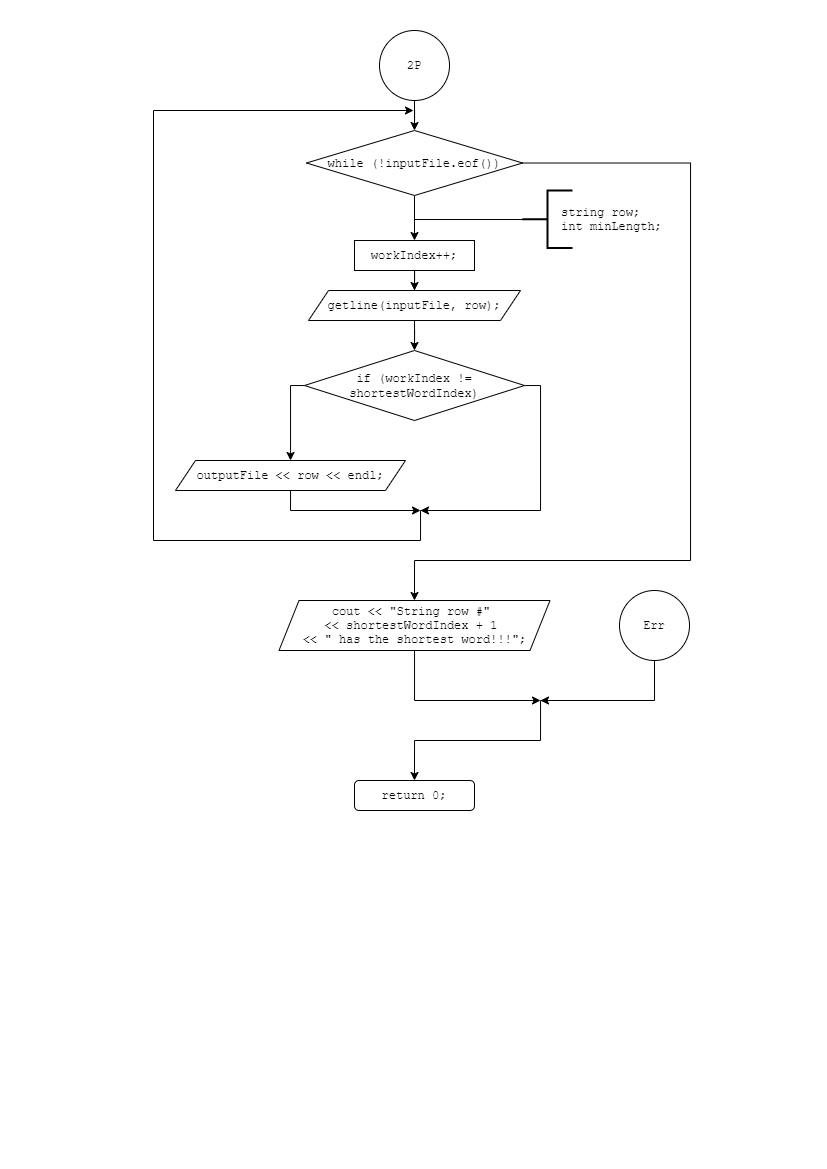
Также, по условию требуется вывести номер строки, содержащей самое короткое слово.

cout << "String row #" << shortestWordIndex + 1

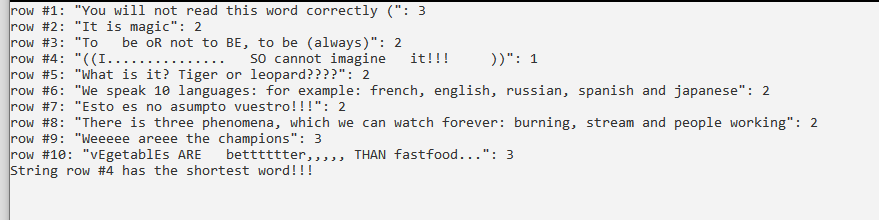
<< " has the shortest word!!!";

Блок-схема решённой задачи:

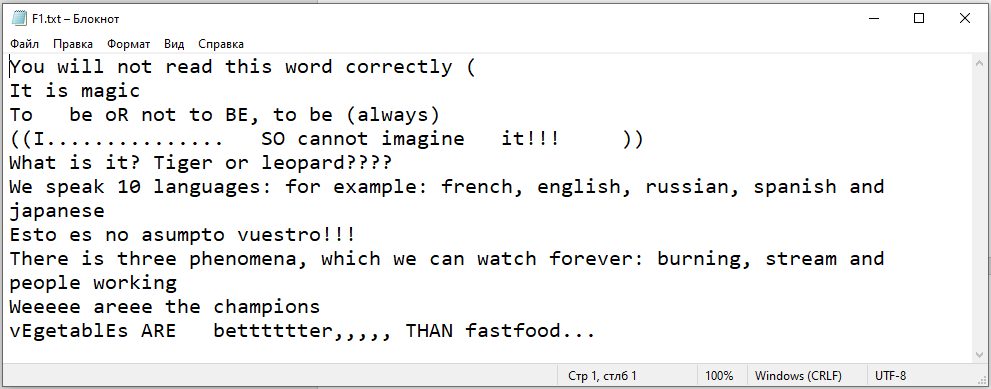




**Скриншот** работы выполненной программы:



Содержимое файла F1.txt



Содержимое файла F2.txt

