Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине «Основы алгоритмизации и программирования»

За 2 семестр

Тема: «Текстовые файлы»

Выполнили:

Студенты 1 курса

Группы ИИ-21(1)

Кабак Д. Н.

Проверил:

Гирель Т. Н.

Брест 2022

**Цель работы:** изучить принципы программирования с использованием индексных файлов.

**Ход работы:**

**Вариант 3**

**Задание 1**

Реализовать меню, разработанное с применением массива указателей на функции.

**Задание 2**

Предварительно подготовить(программно или с помощью текстового редактора) текстовый файл, содержимым которого является массив (матрица) из чисел. В соответствии с индивидуальным заданием написать программу обработки содержимого файла.

Для каждого из вариантов задания необходимо выполнить следующие действия. Просмотреть содержимое исходного файла. Считать содержимое файла в одномерный (двумерный) динамический массив. Обработать

динамический массив согласно варианту задания. Полученный результат записать в конец исходного файла.

Вариант сортировки: сортировка простыми вставками (InsertSort).

**Задание 3** **«Основные операции с текстовыми файлами»**

Дано имя файла и целое число N (0 < N < 27). Создать текстовый файл с указанным именем и записать в него N строк длины N; строка с номером K (K = 1, …, N) должна содержать K начальных прописных (то есть заглавных) латинских букв, дополненных справа символами «\*» (звездочка). Например, для N = 4 файл должен содержать строки «A\*\*\*», «AB\*\*», «ABC\*», «ABCD».

**Задание 4 «Анализ и форматирование текста»**

Дан текстовый файл. Найти количество абзацев в тексте, если первая строка каждого абзаца начинается с 5 пробелов («красная строка»). Пустые строки между абзацами не учитывать.

**Задание 5 «Текстовые файлы с числовой информацией»**

Даны вещественные числа A, B и целое число N. Создать текстовый файл, содержащий таблицу значений функции x на промежутке [A, B] с шагом (B – A)/N. Таблица состоит из двух столбцов: с аргументами x (10 позиций, из них 4 под дробную часть) и со значениями x (15 позиций, из них 8 под дробную часть). Столбцы выравниваются по правому краю.

**Код:**

#include <iostream>

#include <string>

#include <fstream>

#include <cstdlib>

#include <vector>

#include <cmath>

#include <windows.h>

#include <ctime>

using namespace std;

void task1();

void task2();

void task3();

void task4();

void task5();

int menu();

void(\*masf[])() = {task1, task2, task3, task4, task5};

int main() {

    while(1) (\*masf[menu()])();

}

int menu() {

    char n;

    do

    {

        cout << "\t   Menu\n";

        cout << "\t1. Task 1.\n";

        cout << "\t2. Task 2.\n";

        cout << "\t3. Task 3.\n";

        cout << "\t4. Task 4.\n";

        cout << "\t5. Task 5.\n";

        cout << "\t6. Exit.\n";

        printf ("Make a choice: ");

        n = getchar();

        system("cls");

    }

    while(strchr("123456", n) == NULL);

    if(n == '6') exit(0);

    return n - 49;

}

void task1() {

    system("cls");

    cout << "\nYou are already in the first task.\n";

    system("pause");

}

void task2()

{

    system("cls");

    string PATH = "file2\_den.txt";

    ofstream file\_out;

    ifstream file\_in;

    srand(time(NULL));

    file\_out.open(PATH, ios\_base::out);

    for (int i = 0; i < 10; i++) {

        file\_out << int(rand()) << "\t";

    }

    file\_out.close();

    file\_in.open(PATH, ios\_base::in);

    string s, sum\_s = "";

    while (getline(file\_in, s)) {

        sum\_s += s;

    }

    file\_in.close();

    vector<int> arr;

    string split("\t");

    size\_t prev = 0;

    size\_t next;

    size\_t delta = split.length();

    while ((next = sum\_s.find(split, prev)) != string::npos){

        arr.push\_back(atoi(sum\_s.substr(prev, next-prev).c\_str()));

        prev = next + delta;

    }

    arr.push\_back(atoi(sum\_s.substr(prev).c\_str()));

    int buff = 0; // для хранения перемещаемого значения

    int i, j; // для циклов

    for (i = 1; i < arr.size(); i++) {

        buff = arr[i]; // запомним обрабатываемый элемент

        // и начнем перемещение элементов слева от него

        // пока запомненный не окажется меньше чем перемещаемый

        for (j = i - 1; j >= 0 && arr[j] > buff; j--) arr[j + 1] = arr[j];

        arr[j + 1] = buff; // и поставим запомненный на его новое место

    }

    file\_out.open(PATH, ios\_base::app);

    file\_out << endl << endl;

    for (int i = 1; i < arr.size(); i++) {

        file\_out << arr[i] << "\t";

    }

    file\_out.close();

    system("pause");

}

void task3() {

    system("cls");

    string PATH = "file3\_den.txt";

    ofstream file\_out;

    int N;

    while (true) {

        cout << "Enter N: ";

        cin >> N;

        if (N > 0 && N < 27) break;

    }

    string s = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    file\_out.open(PATH, ios\_base::out);

    for (int i = 1; i <= N; i++) {

        for (int j = 0; j < i; j++) {

            file\_out << s[j];

        }

        for (int j = 1; j <= N - i; j++) {

            file\_out << "\*";

        }

        file\_out << endl;

    }

    file\_out.close();

    system("pause");

}

void task4() {

    system("cls");

    string PATH = "file4\_den.txt";

    ifstream filei;

    filei.open(PATH, ios\_base::in);

    filei.seekg(0,ios\_base::end);

    int k = filei.tellg();

    int space = 0, sum\_red\_line = 0;

    for(int i = 0; i < k; i++) {

        filei.seekg(i,ios\_base::beg);

        char buffer[] = "";

        filei.read(buffer, 1);

        if (space == 5) {

            sum\_red\_line += 1;

            space = 0;

        }

        if (buffer[0] == ' ') space += 1;

    }

    cout << "The count of red lines: " << sum\_red\_line;

    filei.close();

    system("pause");

}

void task5() {

    system("cls");

    string PATH = "file5\_den.txt";

    ofstream fileo;

    fileo.open(PATH, ios\_base::out);

    float A, B;

    int N;

    cout << "Enter A: ";

    cin >> A;

    cout << "Enter B: ";

    cin >> B;

    cout << "Enter N: ";

    cin >> N;

    float X = 0;

    for(int i = 0; i < N; i++) {

        X += (B - A) / N;

        fileo.width(10);

        fileo.precision(4);

        fileo << right << X;

        fileo.width(15);

        fileo.precision(8);

        fileo << right << sin(X);

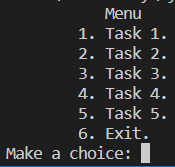
        fileo<<endl;

    }

    fileo.close();

    system("pause");

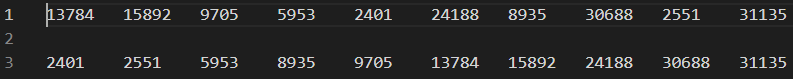
}

****

Задание 1:

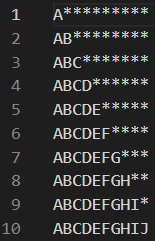
****

Задание 2:

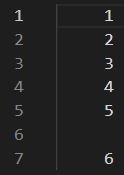


Задание 3:

****

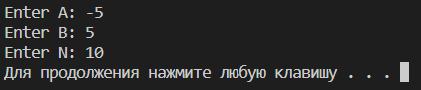
****

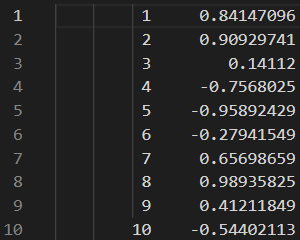
Задание 4:

****

****

Задание 5:

****

****

**Вывод:** изучил принципы программирования с использованием индексных файлов.