МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы №1

по дисциплине “Технологии Программного Обеспечения”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

Проверил: доц. каф.

«Информационные системы»

Строганов В. А.

Севастополь

2022

**ЛАБОРАТОРНАЯ РАБОТА №1**

**«Исследование способов анализа областей эквивалентности и построения тестовых последовательностей»**

**Цель работы**

Исследовать способы анализа областей эквивалентности входных данных для тестирования программного обеспечения. Приобрести практические навыки составления построения тестовых последовательностей.

**Задачи**

Задача 1.

Дана целочисленная квадратная матрица. Определить произведение элементов в тех строках,

которые не содержат отрицательных элементов.

Задача 2.

Дана строка. Преобразовать строку, заменив в ней каждую точку многоточием.

Задача 3.

Программа, которая считывает английский текст из файла и выводит на экран слова,

начинающиеся с гласных букв.

**Ход работы**

**Задача 1**

Для начала выполнения работы была cоставлена программа, выполняющая заданные действия для Задачи 1 (Приложение А).

Затем были определены области эквивалентности.

1)По размеру матрицы:

а) Матрица состоит из одного элемента;

б) Матрица состоит более чем из одного элемента.

2)По наличию и расположению положительных элементов:

А) Все элементы положительные;

б) Существует один отрицательный элемент;

в) Существует несколько отрицательных элементов в 1 строке;

г) Существует отрицательные элементов в разных строках;

д) Все элементы отрицательные.

После этого были осуществлены тестовые примеры выполнения программы для каждой из областей эквивалентности (Рисунки 1-5).

Тестовая последовательность для матрицы из 1 отрицательного элемента (Рисунок 1).

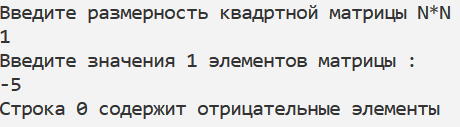


Рисунок 1 – Ввод единственного отрицательного элемента.

Тестовая последовательность для матрицы из 1 элемента (Рисунок 2).

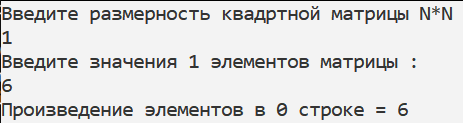


Рисунок 2 – Ввод единственного положительного элемента.

Тестовая последовательность для матрицы из более чем 1 элемента, где все отрицательные (Рисунок 3).

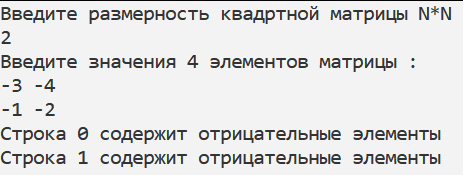


Рисунок 3 – Ввод нескольких всех отрицательных элементов.

Тестовая последовательность для матрицы из более чем 1 элемента, где есть один положительный (Рисунок 4).

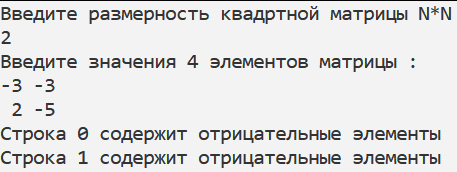


Рисунок 4 – Ввод нескольких элементов с 1 положительным.

Тестовая последовательность для матрицы из нескольких элементов, где есть несколько положительных элементов (Рисунок 5).

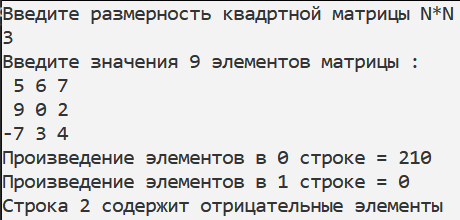


Рисунок 5 – Ввод более 1 элемента с несколькими положительными.

**Задача 2**

Для решения была cоставлена программа, выполняющая заданные действия для Задачи 2 (Приложение Б).

Затем были определены области эквивалентности.

1)По размеру строки:

а) Строка состоит из одного символа;

б) Строка состоит более чем из нескольких символов.

2)По наличию и расположению точек:

А) Нет ни одной точки;

б) Существует один символ точки;

в) Существует несколько точек.

После этого были осуществлены тестовые примеры выполнения программы для каждой из областей эквивалентности (Рисунки 5-10).

Тестовая последовательность для строки из 1 символа не точки (Рисунок 6).

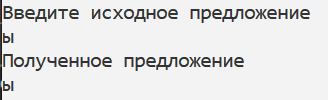


Рисунок 6 – Ввод единственного символа не точку.

Тестовая последовательность для строки из 1 символа, имеющей точку (Рисунок 7).

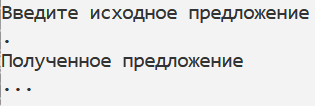


Рисунок 7 – Ввод единственного символа – точки.

Тестовая последовательность для строки из нескольких символов, имеющей ни единой точки (Рисунок 8).

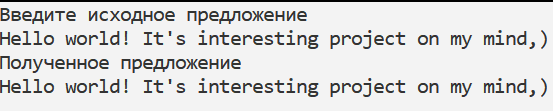


Рисунок 8 – Ввод нескольких символов без единой точки.

Тестовая последовательность для строки из нескольких символов, имеющей один символ точки (Рисунок 9).

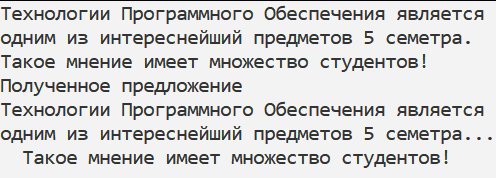


Рисунок 9 – Ввод нескольких символов с одной точкой.

Тестовая последовательность для строки из нескольких символов, имеющей более одного символа точки (Рисунок 10).

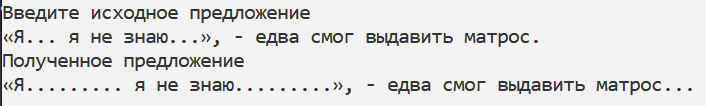


Рисунок 10 – Ввод нескольких символов более чем 1 точкой.

**Задача 3**

Для решения этого была cоставлена программа, выполняющая заданные действия для Задачи 3 (Приложение В).

Затем были определены области эквивалентности.

1)По количеству слов:

а) Текст состоит из одного слова;

б) Текст состоит из нескольких слов.

2)По наличию и расположению английских слов, начинающихся на гласные буквы:

А) Нет ни одного английского слова, начинающего на гласную букву;

б) Существует одно английское слово, начинающееся на гласную букву;

в) Существует несколько английских слов, начинающихся на гласную букву.

Тестовая последовательность для текста из одного слова, не начинающегося на гласную английскую гласную букву (Рисунок 11).

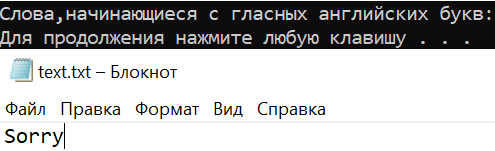


Рисунок 11 – Ввод текста из одного 1 слова, не начинающего на английскую гласную.

Тестовая последовательность для текста из одного слова, начинающегося на гласную английскую гласную букву (Рисунок 12).

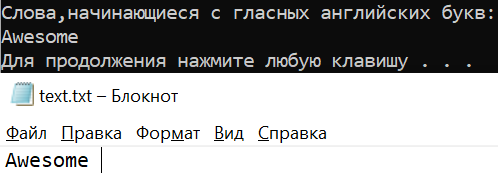


Рисунок 12 – Ввод текста из одного 1 слова, начинающего на английскую гласную.

Тестовая последовательность для текста из нескольких слов, не начинающегося на гласную английскую гласную букву (Рисунок 13).

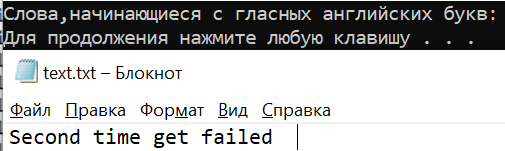


Рисунок 13 – Ввод текста из нескольких слов, не начинающего на английскую гласную.

Тестовая последовательность для текста из нескольких слов, где одно из них начинается на английскую гласную букву (Рисунок 14).

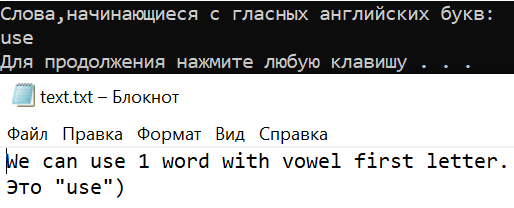


Рисунок 14 – Ввод текста из нескольких слов, где одно начинается на английскую гласную букву.

Тестовая последовательность для текста из нескольких слов, где больше одного из них начинаются на английскую гласную букву (Рисунок 15).

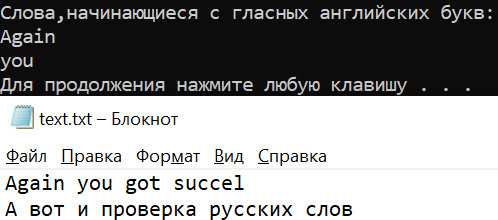


Рисунок 15 – Ввод текста из нескольких слов, где больше одного начинается на английскую гласную букву.

**Выводы**

В ходе лабораторной работы были исследованы способы анализа областей

эквивалентности входных данных для тестирования программного обеспечения, в ходе которого были получены знания о разделении входных данных на классы. Также были приобретены практические навыки составления построения тестовых последовательностей: по количеству входных данных, по их наличию и расположению.

**ПРИЛОЖЕНИЕ А**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "Введите размерность квадртной матрицы N\*N" << endl;

int N;

cin >> N;

int\* ant = new int[N];

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

ant[i] = 1;

int\*\* M = new int\* [N];

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

M[i] = new int[N];

//Ввод значений в матрицу

cout << "Введите значения " << N \* N << " элементов матрицы :\n";

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

for (int j = 0; j < N; j++) {

cin >> M[i][j];

}

// Вычисление

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

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

{

ant[i] \*= M[i][j];

if (M[i][j] < 0) {

ant[i] = -1;

break;

}

}

//Вывод результата на экран

for (int i = 0; i < N; i++) {

if (ant[i] >= 0)

cout << "Произведение элементов в " << i << " строкe = " << ant[i] << endl;

else cout << "Строка " << i << " содержит отрицательные элементы\n";

}

}

**ПРИЛОЖЕНИЕ Б**

#include<iostream>

#include<Windows.h>

#include<string>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n;

string s;

cout << "Введите исходное предложение" << endl;

getline(cin, s);

for (int i = 0; i < s.size(); i++)

{

if (s[i] == '.') {

s.insert(i++, ".");

s.insert(i++, ".");

}

}

cout << "Полученное предложение\n";

cout << s << endl;

return 0;

}

**ПРИЛОЖЕНИЕ В**

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

char slovo[20];

ifstream in("text.txt");

cout << "Слова, начинающиеся с гласных английских букв: \n";

while (!in.eof()) {

in >> slovo;

if (slovo[0] == 'A'

|| slovo[0] == 'a'

|| slovo[0] == 'E'

|| slovo[0] == 'e'

|| slovo[0] == 'Y'

|| slovo[0] == 'y'

|| slovo[0] == 'U'

|| slovo[0] == 'u'

|| slovo[0] == 'I'

|| slovo[0] == 'i'

|| slovo[0] == 'O'

|| slovo[0] == 'o')

cout << slovo << endl;

}

system("pause");

return 0;

}