Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

|  |
| --- |
| Институт информационных технологий и анализа данных |

наименование института

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине:

|  |
| --- |
| Дискретная математика |
| Использование языковых средств программирования для  задач теории множеств. |

обозначение документа

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил | АСУб-20-2 |  |  |  | Арбакова А.В. |
|  | шифр группы |  | подпись |  | Фамилия И.О. |
| Проверил |  |  |  |  | Носырева Л.Л. |
|  | должность |  | подпись |  | Фамилия И.О. |

Иркутск 2021 г.

**Вариант 1**

**2. Цели работы:**

Получение навыков работы с множествами.

**3. Задание:** Дана непустая последовательность слов из строчных русских букв; между соседними словами – запятая, за последним словом – точка. Напечатать множества, сформированные согласно варианту.

**4. Листинг программы:**

#include <iostream>

#include <windows.h>

using namespace std;

int main()

{

char \*delimiters = ", ";

char \*gluh = "кпстфхцчшщ";

char txt[256];

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Введите последовательность слов через запятую ( . - конец ввода): (пример) Арбуз, мандарин, апельсин, гранат. " << endl;

cin.getline(txt, 256);

CharLowerBuffA(txt, strlen(txt));

cout << "Множество букв, входящих в текст не менее 2 раз: ";

for (int i = 0; txt[i]!=0&& txt[i] != '.'; i++)

{

if ((txt[i] >= 'а'&&txt[i] <= 'я') || txt[i] == 'ё')

{

int cnt\_let = 0;

for (int j = 0; txt[j] != 0 && txt[j] != '.'; j++)

{

if (txt[i] == txt[j])

{

if (j < i)

break;

cnt\_let++;

}

}

if (cnt\_let >= 2)

cout << " " << txt[i];

}

}

cout << endl;

cout << "Множество глухих согласных букв, которые входят в каждое нечетное слово и не входят хотя бы в одно четное слово: ";

for (int k = 0; gluh[k]!=0; k++)

{

int nw = 0;

bool alleven = true;

bool allodd = true;

for (int i = 0; txt[i] != 0 && txt[i] != '.';)

{

//ищем начало слова

for (; txt[i] != 0 && txt[i] != '.' && strchr(delimiters, txt[i]); i++);

if (txt[i] != 0 && txt[i] != '.')

{

nw++;

bool is\_in\_word = false;

for (; txt[i] != 0 && txt[i] != '.' && strchr(delimiters, txt[i]) == NULL; i++)

{

if (txt[i] == gluh[k])

{

is\_in\_word = true;

}

}

if (nw % 2 == 1)

{

if (!is\_in\_word)

allodd = false;

}

else

{

if (!is\_in\_word)

alleven = false;

}

}

}

if (allodd && !alleven)

cout << " " << gluh[k];

}

cout << endl;

system("pause");

return 0;

}

**5.Таблица тестов:**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Результат работы программы | |
| Множество всех букв, входящих в текст не менее 2 раз. | Все глухие согласные буквы, которые входят в каждое нечетное слово и не входят хотя бы в одно четное слово. |
| 1 | Кошка, черепаха, кит, слон, кот. | к о а е т | к |
| 2 | Арбуз, мандарин, апельсин, гранат. | а р н и |  |
| 3 | Ромашка, лилия, гвоздика, роза. | р о а к л и з | к |
| 4 | Помидор, огурец, петрушка, морковь, перец. | п о м р у е ц к | п |