Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 13

По дисциплине «Основы алгоритмизации и программирования»

на тему «**Обработка символьной информации**»

Выполнила:

Студент(ка) 1 курса 7 группы

Гриценко Анна Александровна

Преподаватель: асс. Андронова М.В.

2023, Минск

**Дополнительные задания**

1.В заданной последовательности слов найти все слова, имеющие заданное окончание.

#include<iostream>// Подключаем библиотеку для работы с вводом(выводом)

#include<Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_CTYPE, "Rus");

const int StrSize = 300;//Инициализируем константу

int wordLength{}, sizeResultStr{};//Инициализируеи переменные

char str1[StrSize]{}, resultStr[StrSize]{}, endOfTheWord;//Инициализируем массивы

//Вводим с клавиатуры строку

cout << "Введите строку: ";

gets\_s(str1);//считывание строки из стандартного ввода

//Вводим с клавиатуры окончание слова

cout << "Введите окончание слова: ";

cin >> endOfTheWord;

for (int i = 0; str1[i] != '\0'; i++)//с помощью цикла ищем слова с заданным окончанием

{

++wordLength;

if (str1[i + 1] == ' ' && str1[i] == endOfTheWord)

{

i -= (wordLength - 1);

wordLength += i;

for (; i != wordLength; i++)

{

resultStr[sizeResultStr] = str1[i];

++sizeResultStr;

}

resultStr[sizeResultStr] = ' ';

++sizeResultStr;

wordLength = 0;

continue;

}

if (str1[i + 1] == '\0' && str1[i] == endOfTheWord)

{

i -= (wordLength - 1);

wordLength += i;

for (; i != wordLength; i++)

{

resultStr[sizeResultStr] = str1[i];

++sizeResultStr;

}

resultStr[sizeResultStr] = ' ';

++sizeResultStr;

wordLength = 0;

break;

}

if (str1[i] == ' ')

{

wordLength = 0;

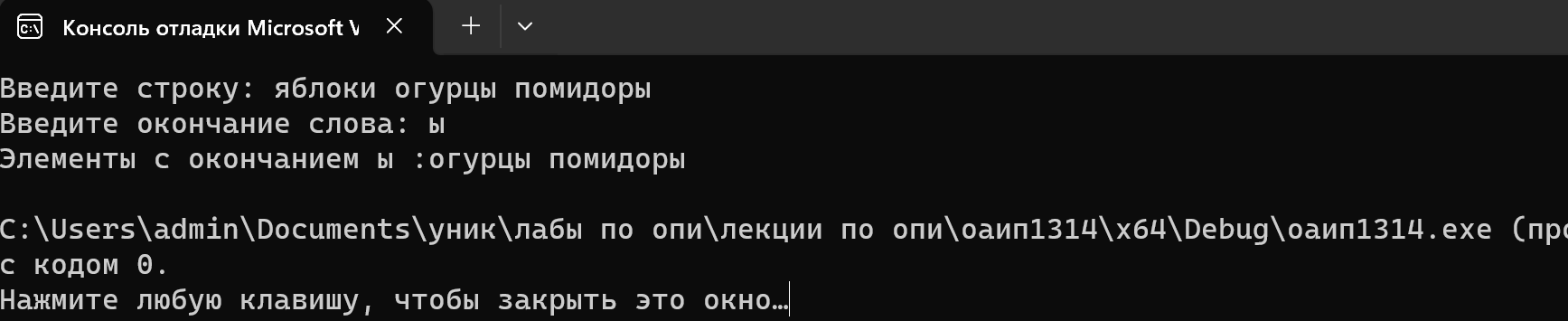
}

}

cout << "Элементы с окончанием "<< endOfTheWord<< " :" << resultStr << endl;

return 0;

}



4. Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами. Пример: HOW DO YOU DO преобразовать в OD OD.

#include<iostream>// Подключаем библиотеку для работы с вводом(выводом)

#include<Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_CTYPE, "Rus");

const int StrSize = 300;//Инициализируем константу

int wordNumber = 1, size2{}, wordlenght;

char string1[StrSize]{}, string2[StrSize]{};//Инициализируем массивы

//Вводим с клавиатуры строку

cout << "Введите строку: ";

gets\_s(string1);//считывание строки из стандартного ввода

for (int i = 0; string1[i] != '\0'; i++)

{

if(string1[i] == ' ')

{

++wordNumber;

}

if (wordNumber % 2 == 1)

{

string1[i] = ' ';

}

}

//с помощью цикла убираем лишние пробелы

for (int i = 0; string1[i] != '\0'; i++)

{

if (string1[i] == ' ' && string2[size2 - 1] == ' ')

{

continue;

}

string2[size2] = string1[i];

++size2;

}

//с помощью цикла освобождаем массив

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

{

string1[i] = '\0';

}

//с помощью цикла делаем результат преобразований массива

for (int i = 0; i < size2; i++)

{

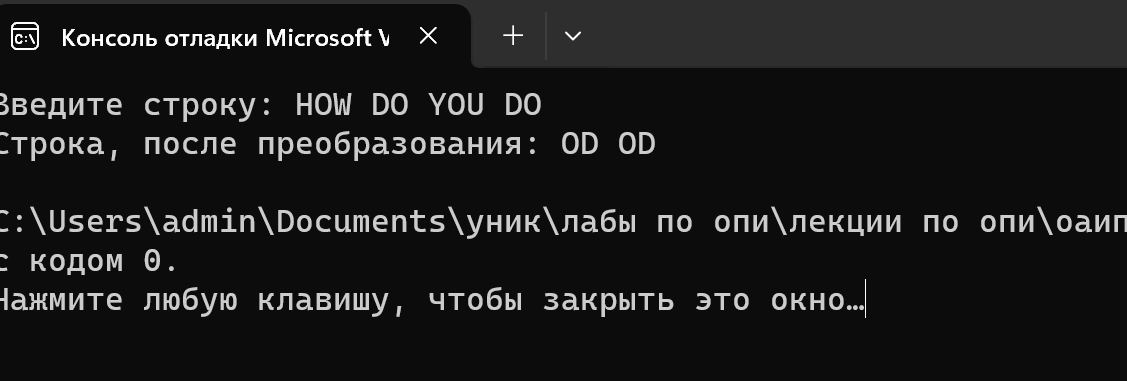
string1[i] = string2[size2 - i - 1];

}

cout << "Строка, после преобразования: "<< string1 << endl;

return 0;

}

****

3.Из заданного предложения удалить те слова, которые уже встречались в предложении раньше.

#include<iostream>// Подключаем библиотеку для работы с вводом(выводом)

#include<Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_CTYPE, "Rus");

const int StrSize = 300, wordCount = 30, numberOfLetters = 30;//Инициализируем константу

int word{}, letters{}, sizeResultOffer{};

char string1[StrSize]{}, string2[StrSize]{}, variousWords[wordCount][numberOfLetters]{};//Инициализируем массивы

//Вводим с клавиатуры строку

cout << "Введите строку: ";

gets\_s(string1);//считывание строки из стандартного ввода

//добавляем между словами пробелы

for (int i = 0; string1[i] != '\0'; i++)

{

if (string1[i] == ' ')

{

++word;

letters = 0;

continue;

}

variousWords[word][letters] = string1[i];

++letters;

}

//с помощью цикла ищем одинаковые слова

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

{

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

{

if (i == j)

{

continue;

}

if (variousWords[i][0] == variousWords[j][0])

{

for (int g = 1; variousWords[i][g] != '\0'; g++)

{

if (variousWords[i][g] == variousWords[j][g])

{

variousWords[j][0] = ' ';

continue;

}

}

}

}

}

//Формируем конечный массив элементов

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

{

for (int j = 0; variousWords[i][j] != '\0'; j++)

{

if (variousWords[i][0] == ' ')

{

break;

}

string2[sizeResultOffer] = variousWords[i][j];

++sizeResultOffer;

}

if (variousWords[i][0] != ' ')

{

string2[sizeResultOffer] = ' ';

++sizeResultOffer;

}

}

cout << "Строка, после преобразования: "<< string2 << endl;

return 0;

}

