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:

Студентка 1 курса 10 группы

Рублевская Маргарита Владимировна

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант№ 13**

1. Выполнить задания из таблицы ниже, используя ***индексы*** для доступа к элементам массивов в первой программе и ***указатели*** − во второй программе. При написании программне использовать стандартные функции для строк символов.

1. Отредактировать заданное предложение, удаляя из него все слова с чётными номерами

Код:

#include <stdio.h> // Подключаем заголовочный файл, обеспечивающий выполнение операций ввода/вывода

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

int main()

{

using namespace std;

setlocale(LC\_CTYPE, "Russian");

char predl[1000], nov[1000]; // Объявляем массив

int i, u = 0, nom; //Ввод переменных

printf("Введите предожение: ");

gets\_s(predl); // Функция принимает в качестве параметра адрес массива символов и считывает строку из стандартного потока, помещая её в данный массив

nom = 1;

for (i = 0; predl[i] != 0; i++)

{

if (predl[i] == ' ')

{

nom++;

if (nom % 2 == 0)

{

i++;

for (i; predl[i] != ' ' && predl[i] != '\0'; i++);

i--;

}

else

{

nov[u] = predl[i];

u++;

}

}

else

{

nov[u] = predl[i];

u++;

}

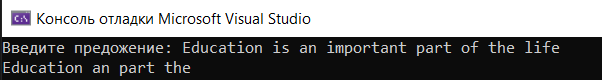
}

for (i = 0; i < u; i++)

printf("%c", nov[i]);

}

Консоль VS:



2. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Определить порядковый номер слова, один из символов которого находится на **k**-й позиции (если на **k**-ю позицию попадает пробел, то найти номер предыдущего слова).

Код:

#include <stdio.h>

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

int main()

{

using namespace std;

setlocale(LC\_CTYPE, "Russian");

char predl[1000]; // Объявляем массив

int i, k, nom; //Ввод переменных

cout << "Введите предожение: ";

gets\_s(predl);

cout << "Введите позицию символа k: ";

cin >> k;

nom = 1;

// Определяем порядковый номер слова, один из символов которого находится на **k**-й позиции

for (i = 0; i <= k - 1; i++) // Отсекаем 0 символ

{

if (\*(predl + i) == ' ')

nom++;

}

if (\*(predl + k - 1) == ' ')

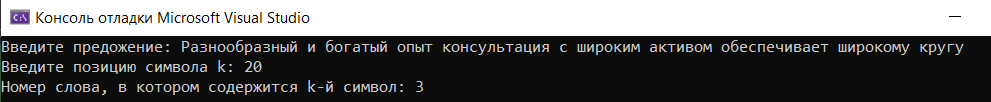
cout << "Номер предыдущего слова: " << nom - 1 << endl;

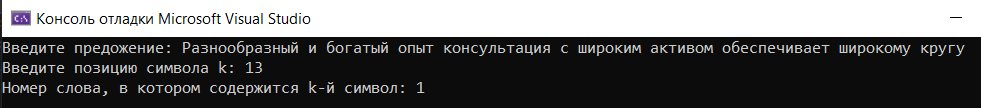
else

cout << "Номер слова, в котором содержится k-й символ: " << nom << endl;

}

Консоль VS:





**Дополнительные задания**

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

Код:

#include <stdio.h>

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

int main()

{

using namespace std;

setlocale(LC\_ALL, "Russian");

char slova[100], okonch[10], v\_slova[100]; // Объявляем массив

int i, k, j = 0, poz = 0; //Ввод переменных

bool a;

printf("Введите слова: ");

fgets(slova, 100, stdin);

printf("Введите окончание: ");

fgets(okonch, 10, stdin);

// Осуществляем обработку строки slova и создает новую строку v\_slova, содержащую определенную последовательность символов.

for (i = 0; i <= strlen(slova); i++)

{

if (slova[i] == ' ' || slova[i] == '\0')

{

poz = i;

if (slova[i] != '\0') {

i = i - strlen(okonch) + 1;

}

else {

i = i - strlen(okonch);

}

a = true;

for (i, k = 0; i < poz; i++, k++)

{

if (slova[i] != okonch[k])

{

a = false;

break;

}

}

if (a == true)

{

for (i = poz - 1; i != -1 && slova[i] != ' '; i--) {

}

i++;

for (i, j; i < poz; i++, j++) {

v\_slova[j] = slova[i];

}

v\_slova[j] = ' ';

j++;

}

i = poz + 1;

}

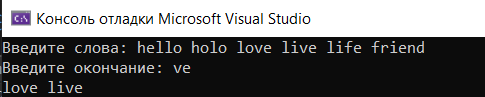
}

for (i = 0; i < j - 1; i++)

printf("%c", v\_slova[i]);

}

Консоль VS:



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

Код:

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

int main() {

setlocale(LC\_CTYPE, "Russian");

const int maxSize = 1000; // Объявляем константу\

//Ввод переменных

char sentence[maxSize];

char result[maxSize];

int i, u = 0, wordCount;

std::cout << "Введите предложение: ";

std::cin.getline(sentence, maxSize);

wordCount = 1;

// Считаем количество слов в предложении и формируем новую строку, исключая каждое второе слово.

for (i = 0; sentence[i] != '\0'; i++) {

if (sentence[i] == ' ') {

wordCount++;

if (wordCount % 2 == 0) {

i++;

for (i; sentence[i] != ' ' && sentence[i] != '\0'; i++);

i--;

}

else {

result[u] = sentence[i];

u++;

}

}

else {

result[u] = sentence[i];

u++;

}

}

for (i = 0; i < u; i++) {

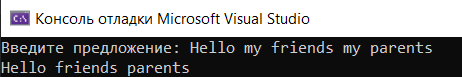
std::cout << result[i];

}

return 0;

}

Консоль VS:



3. 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.

Код:

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

#include <string>

#include <sstream>

int main() {

setlocale(LC\_CTYPE, "Russian");

std::string text; // Строка для хранения введенного текста

std::cout << "Введите предложение: " ;

std::getline(std::cin, text); // Считываем строки текста

std::istringstream iss(text); // Создаем объект istringstream для разбиения строки на слова

std::string word;

int count = 1;

while (iss >> word) {

if (count % 2 == 0) { // Если порядковый номер слова четный

for (int i = word.length() - 1; i >= 0; i--) { // Переворачиваем слово

std::cout << word[i];

}

std::cout << " ";

}

count++;

}

std::cout << std::endl;

return 0;

}

Консоль VS:

