Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 13

По дисциплине «Основы алгоритмизации и программирования»

На тему «Обработкасимвольнойинформации»

 Выполнила:

Студентка1 курса 6 группы

Альшевская Алина Михайловна

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 1

Задание 1

**Написать программу, реализующую вставку в строку n символов, начиная с позиции k.**

#include <cstring>

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "rus");

char str[100],s;

int n, k;

cout << "Введите строку : ";

cin >> str;

cout << "Количесво символов : ";

cin >>n;

cout << "Символ : ";

cin >> s;

cout << "С какой позиции : ";

cin >> k;

cout << "Результат : ";

for (int i = 0; i < k; i++) {//выводим строку до позиции k

cout << str[i];

}

for (int i = 0; i < n; i++) {//выводим n символов s

cout << s;

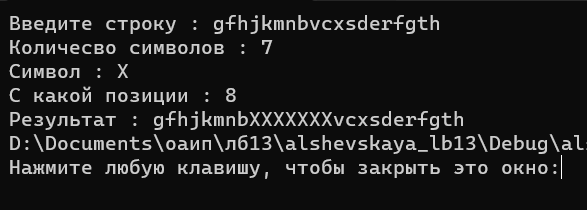
}

for (int i = k; i < strlen(str); i++) {//заканчиваем выводить строку с позиции к до конца

cout << str[i];

}

}

****

Задание 2

Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется от другой одним или несколькими пробелами. Найти количество групп с пятью символами.

#include <cstring>

#include <string>

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "rus");

char str[100];

char s = ' ';

int kol=0,counter=0;

char \*ps = &s;

cout << "Введите строку : ";

cin.getline(str, sizeof(str));//вводим строку

for (int i=0; i < 100; i++) {//идём по строке и сверяем каждый символ с прбелом или конца строки и считаем количесво символов между пробелами

if (\*(str + i) == \*ps ||\*(str+i)=='\0') {

if (kol == 5) {// если количесво символов между пробелами =5 то увеличиваем счетчик на 1

counter++;

}

kol = 0;

}

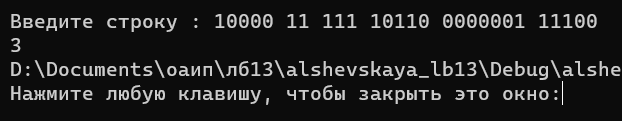
else

kol++;

}

cout << counter;// выводим счётчик

}



Дополнительные задание

Задание 1

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

#include <iostream>

#include <string>

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

const int N = 3;

char arr[N][20] = { "Apple", "Door", "Orange" };

char post[10]; //Переменная в которую записываем окончание

int raw = 0; //Количество совпадений

cout << "Введите окончание: "; cin >> post;

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

int k = strlen(arr[i]) - 1;//последний символ текущего слова

int countPost = strlen(post) - 1; //последний символ окончания

for (k,countPost; countPost >= 0; k--, countPost--) {// идем по циклу пока не дойдём до первого символа окончания

if (arr[i][k] == post[countPost]) {//Сверяем символ слова с символом окочания, если одинаковые, то добавляем совпадения

raw++;

};

};

if (raw == strlen(post)) {//Если количество совпадений и количество символов в окончании равны, то выводим это слово

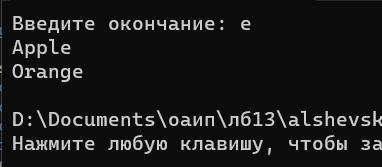
cout << arr[i] << endl;

}

raw = 0;//обнуляем количесво совпадений

}

}



Задание 3

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

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "iostream"

#include "conio.h"

using namespace std;

char\* del(char\* inStr)

{

char\* newStr = new char[strlen(inStr) + 1]; // Итоговая (результирующая) строка

char\* bufStr = new char[strlen(inStr) + 1]; // Буфер обмена (хранение текущего слова)

\*newStr = '\0'; // Обнуляем результирующую строку newStr

int i = 0; // Индекс текщей позиции в буфере

while (true)

{

if (\*inStr == ' ' || \*inStr == '\0')

{

bufStr[i] = '\0'; // буферу обмена присваеваем симбол '\0'

if (strstr(newStr, bufStr) == NULL) // Ищем в результирующей строке слово расположенное в буфере

{

strcat(newStr, bufStr); // Добовляем в конец результирующей строки слово из буфера

}

while (\*inStr == ' ') // Пропускаем все пробелы и запятые

{

int len = strlen(newStr);

newStr[len] = \*inStr++; // Добавляем эти символы в результирующую строку

newStr[len + 1] = '\0'; // переходим в конец

}

i = 0; // Заново забиваем буфер

}

if (\*inStr == '\0') //Выходим из предложения

{

break;

}

bufStr[i++] = \*inStr++; // Копируем текущей символ с входной строки в буфер

}

delete bufStr;

return newStr;

}

int main()

{

char\* t = new char[255];

cin.getline(t, 255); // Вводим строку

char\* s = new char[strlen(t) + 1]; // выделяем новую память под размер введённой строки

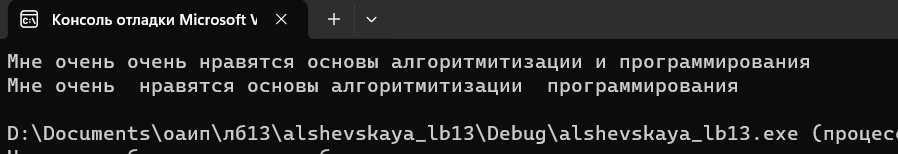
strcpy(s, t); // копируем строку в новую память

delete[] t; // ненужную память освобождаем

s = del(s);

cout << s << "\n";

return 0;

}

Задание 5

Даны два предложения. Найти самое короткое из слов первого предложения, которого нет во втором предложении.

#include<iostream>

#include <algorithm>

#include <unordered\_set>

#include <iterator>

#include <sstream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "rus");

const string s1{ "Мне очень нравятся лекции по основам аргоритмитизации и программирования" };//объявляем константные строки и заполняем содержимым

const string s2{ "Я люблю лекции по основам алгоритмитизации и программирования" };

cout << s1<<"\n";//выводим строки на экран

cout << s2<<"\n";

stringstream sss(s2);//Создаем строковый поток sss и заполняет его содержимым строки s2

unordered\_set<string>sal{ istream\_iterator<string>(sss),istream\_iterator<string>() };//Создаем неупорядоченное множество sal и заполняет его словами из потока sss

sss.str("");

sss.clear();//Очищаем поток sss и заполняем его содержимым строки s1

sss << s1;

cout << \*min\_element(istream\_iterator<string>(sss), istream\_iterator<string>(), [&sal](const string& a, const string& b) {//Выводим на экран минимальное по длине слово из потока sss, которого нет в множестве sal

return a.length() < b.length() && sal.find(a) == sal.end();

});

return 0;

}