**Лабораторна робота №7.Написання програм з використанням символьних та рядкових величин**

**На період** карантину в дистанційній формі навчання потрібно використовувати замість **С++ Builder онлайн компілятор C++ Shell, який доступний за адресом** [**http://cpp.sh**](http://cpp.sh) (інструкція надана після тексту ЛР№3). При наявності на домашньому комп’ютері іншого компілятору С++ завдання можна виконати на ньому. Результати викласти на платформу коледжу в свій репозиторій та надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді cpp-файлу з іменем у форматі

**<Номер групи><Номер лабораторної>[-<Номер завдання>] <Прізвищеанглійською>**

Наприклад,21-1Loban.cpp.

При відсутності можливості доступу до Інтернету текст програми набрати в Блокноті або WordPad Windows та надіслати на електронну адресу викладача

**Строк відсилки ЛР 03.05.2020.**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача.

**Мета**: навчитися писати програми з використанням символьних та рядкових величин, структур та масивів **на мові С++** в **консольному режимі.**

**Методичні вказівки щодо організації самостійної роботи студентів**

1. Прочитати (повторити) основні теоретичні відомості. Переглянути лекцію №9 "Символьні та рядкові величини. Складні типи даних – масиви, структури. приклади програм з ПР№7", а також розгляд прикладів програм з символьними та рядковими величинами в практичній роботі №7. "Розгляд прикладів програм з символьними та рядковими величинами, масивами та структурами".
2. Запустити середовище програмування С++ .
3. Записати програму, що виконує 3 завдання з пп.4.1-4.3 (за завдання 4.1 мінімальна оцінка 2 бали, за 4.2 та 4.3 будуть надані додаткові бали, по 3 за кожне). В першому рядку кожної програми записати

*// ПТБД-21 Група № Прізвище*

вказавши номер своєї групи та своє прізвище.

Як і в ЛР №5, програма повинна запитати номер завдання (число 1, 2, 3 або 0 для закінчення) і в залежності від введеного значення виконувати відповідне завдання. Якщо введений 0 – програма припиняє роботу.

Вхідні дані ввести, а результати вивести, використовуючи потокове введення-виведення даних.

1. Завдання:
   1. Написати програму, яка питає ім`я, порівнює з тими, що вона має як елементи символьного масиву (рядки типу string) та вітає або повідомляє, що “не знайома”.

Елементи символьного масиву задати через ініціалізацію (див. приклад в кінці теоретичних відомостей).

* 1. В заданому рядку "Don’t cut the bough you are sitting on." знайти і вивести найкоротше і найдовше слово, вказати номери позицій, з яких вони починаються. ***Вказівки:*** перегляньте потрібні функції (пошук символу, виділення підрядка, знаходження довжини рядкової змінної. Вам може буди потрібно ввести проміжні рядкові зміні, а для організації перегляду в циклі булеву змінну. Продумайте алгоритм.
  2. Розробити програму, яка запитує текстовий рядок, і в текстовому рядку замінює будь-яку кількість однакових символів, що йдуть один за одним підряд на один такий же символ та цифру, яка відповідає кількості видалених символів (Наприклад: ‘*1CABk3KKK111DeFf0100fk0cccccc*’=*’1CABk3K212DeFf0101fk0с5*’). ***Вказівки:*** початковий рядок вводити з клавіатури; отриманий рядок виводити під початковим рядком.

**Основні теоретичні відомості**

Рядок – це послідовність символів, яка закінчується нульовим байтом *‘\0’*. Дана послідовність символів записується в подвійних лапках.

Послідовності в С++ представляються масивами або покажчиками, тому стандартно програми мовою C++ зберігають рядки як масиви типу *char*. Для оголошення строкової змінної необхідно всередині програми оголосити масив типу *char* з кількістю елементів, достатньою для зберігання необхідного рядка символів.

Головна відмінність між символьними рядками та іншими типами масивів полягає в тому, що C++ визначає останній елемент масиву. Програми мовою C++ представляють кінець символьного рядка за допомогою символу NULL, який в C++ зображується як спеціальний символ '\0', або нульовий байт.

Створення рядків, наприклад, відбувається наступним чином:

char str2[10]="Hello";

Для стандартної роботи з символьними рядками в стилі С++ призначена бібліотека string.h, яка містить наступні основні функції:

– **char \**strcat*(char \*dest, const char \*src);** – дописує рядок *src* в кінець *dest*;

– **char \**strncat*(char \*dest, const char \*src, size\_t n);** – дописує не більше *n* початкових символів рядка *src* (або весь *src*, якщо його довжина менше) у кінець *dest*;

– **char \**strchr*(const char \*str, int ch);** – шукає символ *ch* у рядку *str*, починаючи з голови, і повертає його адресу, або NULL якщо символ не знайдено;

– **char \**strrchr*(const char \*str, int ch);** – шукає символ *ch* у рядку *str*, починаючи з хвоста і повертає його адресу, або NULL якщо символ не знайдено;

– **int *strcmp*(const char \*str1, const char \*str2);** – лексикографічне порівняння рядків;

– **int *strncmp*(const char \*str1, const char \*str2, size\_t n);** – лексикографічне порівняння перших *n* байтів рядків;

**– char \**strcpy*(char \*dest, const char \*src);** – копіює рядок з *src* у *dest*;

– **char \**strncpy*(char \*dest, const char \*src, size\_t n);** – копіює до *n* байт рядку з *src* у *dest*;

– **size\_t *strlen*(const char \*str);** – повертає довжину рядка;

– **size\_t *strspn*(const char \*str, const char \*alp);** – визначає максимальну довжину початкового підрядка з *str*, що складається виключно з байтів, перерахованих в *alp*;

– **char \**strpbrk*(const char \*str, const char \*alp);** – знаходить перше входження будь-якого символу, перерахованого в *alp*, у *str*;

– **char \**strstr*(const char \*str, const char \*pstr);** – знаходить перше входження рядка *pstr* у *str*.

Однак окрім розглянутого вище стандартного способу роботи з символьними рядками в бібліотеці STL С++ є ще одна бібліотека для роботи з рядками. Вона дозволяє легко вирішувати наступні задачі з високою ефективністю:

– створювати, присвоювати, копіювати і видаляти рядки;

– виконувати перетворення типів символьних змінних;

– порівнювати рядки;

– поєднувати рядки;

– визначати довжину рядка;

– знаходити і заміщати потрібний фрагмент у рядку.

Для використання даної бібліотеки необхідно підключити заголовний файл <string>. Після цього операція створення нового рядка виявиться настільки ж простою, як і створення змінної будь-якого базового типу.

**string** hi(“hello”);

**string** lo=“greetings”;

**string** es=””;

Рядковим змінним можна присвоювати значення, як і змінним будь-яких інших типів:

**string** name(“Fred”);

name = “Paul”;

При цьому всі операції розподілу пам'яті будуть виконані коректно.

Операції + (конкатенація), операція порівняння (==, > тощо) дозволяють легко виконувати відповідні дії над рядками.

**Приклад програми з використанням масиву рядкових величин.**

**#include <iostream>**

**#include <string>**

**#include <Windows.h>**

**using namespace std;**

**//-------------------------------------------------------**

**int main()**

**{**

**system("color F0");**

**string month[5]={"april","may","june","july","august"};**

**string name;**

**bool found=false;**

**cout << "What month? ";**

**cin >> name;**

**for (int i=0;i<5;i++)**

**if (name==month[i])**

**{cout << "Hello, " << name << "!\n";**

**found=true;}**

**if (!found ) cout << "Not found " <<endl;**

**system("pause");**

**return 0;**

**}**

**Результат роботи програми**

