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

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

**Завдання.**

1. Прочитати (повторити) основні теоретичні відомості. Переглянути лекцію №9 " Символьні та рядкові величини. Використання рядків типу string
2. Запустити середовище програмування С++ . Записати програму, що виконує 2 завдання з пп.3.1-3.2. В першому рядку кожної програми записати

*// Група № Прізвище*

вказавши номер своєї групи та своє прізвище

1. В головній програмі скласти командний код подання меню та виклику функцій, в яких надається код, розроблений за пп.3.1-3.2. Як і в ПР №5, програма повинна запитати номер завдання (число 1, 2 або 0 для закінчення) і в залежності від введеного значення виконувати відповідне завдання. Якщо введений 0 – програма припиняє роботу. В програмі використати потокове введення та виведення даних.
   1. Написати функцію, 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).

3.2. 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.

1. Результати 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><Прізвище англійською>**

Наприклад, 31-01Ivanov.cpp.

Іншим рішенням є надсилання поштою посилання на текст програми за URL адреси, яку надає C++Shell, вказавши в темі листа, номер групи прізвище студента та номер ПР.

В темі листа вказати, номер групи, прізвище студента та номер ПР як "ПР№7".

**Строк відсилки ПР для МІВТ/ЕТ-41 10.11.2020**

**МНТ/ЕТ- 41 10.11.2020.**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, В темі листа вказати, номер групи, прізвище студента та номер ПР як "ПР№7" В темі листа вказати, номер групи, прізвище студента, номер ПР та фразу

**Теоретичні відомості.**

Для обробки символьних типів даних бібліотека функцій **string.h** має велику кількість вбудованих функцій, що збільшують продуктивність праці програмістів та скорочують час на розробку програм, наприклад:

* функції перевірки рядків;
* функції маніпулювання рядками.

Функції наводяться у вигляді списків, що згруповані за їх розташуванням у заголовних файлах. Найчастіше надаються прототипи функцій, що описують, як слід використовувати функції у програмах.

Розглянемо ще деякі важливі функції використання рядків типу string, потрібні для цієї практичної роботи.

***Функція присвоювання усього рядка або його частини іншому рядку:***

* **str.assign(st)** — присвоює весь рядок **st** типу **string** або масив **char[ ]** — рядку **str**типу **string;**
* **str.assign(st,k,n)** — присвоює **n** символів рядка **st** рядку **str,** починаючи з **k**-ої позиції.

Ці функції можна використовувати для перетворення рядка типу **char** у рядок **string**. Наприклад:  
*//————————– присваивание строк*  
**char st[ ]=”Пришло жаркое лето!”;**  
**string str;**  
**cout<<” str = “<< str.assign(st,7,12)<<endl;**

Результат виконання:  
**str = жаркое лето!**або  
**string str, st=”O, лето красное, любил бы я тебя!”;  
cout<<“str = “<< str.assign(st,3,12) << endl;**

Результат виконання:  
**str = лето красное.**

***Функція порівняння рядків або їхніх частин:***

* **str.compare(st)** — порівнює рядки **st** та **str** і повертає значення:

**<0 — якщо st < str;**  
**=0 — якщо st = str;**  
**>0 — якщо st > str;;**

* **str.compare(st,k,n)** — порівнює n символів рядка **st** з рядком **str,** починаючи з **k**-ої позиції. Наприклад:

Оскільки тип string є класом, то, щоб порівняти два рядки між собою можна використати операцію **‘= =’**. Якщо два рядки однакові, то результат порівняння буде true. У противному випадку, результат порівняння буде false.

Але, якщо потрібно порівняти частину одного рядка з іншим, то для цього передбачена функція compare().

Функція працює наступним чином. Якщо викликаючий рядок менше рядка s, то функція повертає -1 (від’ємне значення). Якщо викликаючий рядок більше рядка s, функція повертає 1 (додатне значення). Якщо два рядка рівні, функція повертає 0.

**Приклад**. Демонстрація роботи функції compare():

// тип string, функція compare()

string s1 = "012345";

string s2 = "0123456789";

int res;

res = s1.compare(s2); // res = -1

res = s1.compare("33333"); // res = -1

res = s1.compare("012345"); // res = 0

res = s1.compare("345"); // res = -1

res = s1.compare(0, 5, s2); // res = -1

res = s2.compare(0, 5, s1); // res = -1

res = s1.compare(0, 5, "012345"); // res = -1

res = s2.compare(s1); // res = 1

res = s2.compare("456"); // res = -1

res = s2.compare("000000"); // res = 1

**Розглянемо приклади використання наведених функцій при обробці рядків типу string.**

***Приклад 1.*** Вести до пам’яті комп’ютера будь-який текст, відокремити в ньому всі слова, вивести їх на екран та визначити найдовше слово

**#include <iostream>**

**#include <string>**

**using namespace std;**

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

**int main()**

**{**

**string txt, sl, slmax; // slnax – найдовше слово**

**int i, k=0, n=0, max=0;**

**//---------------- Введення тексту**

**cout <<"\*\*\*\*\* Enter text\n";**

**getline(cin,txt,'\n');**

**for (i = 0; i <= txt.length()-1; i++)**

**{ if ((txt[i]==' ') || (txt[i]==',') || (txt[i]=='.'))**

**{n++; // n —лічильник пропусків та знаків пунктуації**

**if (n>1) continue; k++; //k — лічильник слів**

**cout<< k <<" word - "<< sl<<" = "<<sl.size() << " symbols\n";**

**//--------------- визначення слова з найбільшою кількістю літер**

**if (sl.size() > max)**

**{ max = sl.size();**

**slmax=sl; }**

**sl=""; //очищення sl для нового слова**

**}**

**else {n=0; sl=sl+txt[i]; } //змінна s1 — формує слово**

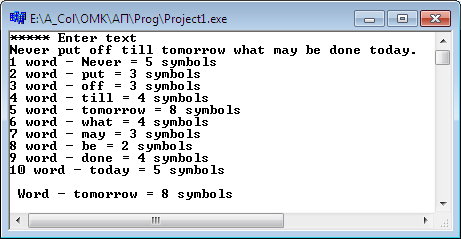
**}**

**cout<<"\n Word - "<< slmax << " = " <<slmax.size()<<" symbols\n";**

**return 0;**

**}**

Результат:



**Приклад 2.** З уведеного списку прізвищ (без ініціалів) вибрати такі, що починаються на задані літери і мають задане закінчення, та вивести повідомлення про прізвище з найменшою кількістю літер.

**#include <iostream>**

**#include <string>**

**#include <conio.h>**

**#include <Windows.h>**

**using namespace std;**

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

**int main()**

**{**

**system("color F0"); // Встановлення кольору фону для С+Builder**

**const int n = 6;**

**string spis [n], pok, p ; /\* Список прізвищ, задані закінчення та перші літери \*/**

**int i, minfam, k=0; /\* додаткові змінні для збереження проміжних результатів \*/**

**int lpok, lspis, lp; //**

**//------------------------------ Введення списку прізвищ**

**cout << "\*\*\*\*\* Enter "<<n<<" name";**

**for (i = 0; i < n; i++)**

**{ cout << "\n" << i+1 << '\t';**

**cin >> spis[i]; }**

**//------------------------------ Введення перших літер**

**cout << "\*\*\*\*\* Enter first letter\n";**

**cin >> p;**

**lp = p.size(); // визначення довжини перших літер**

**//------------------------------ Введення закінчення**

**cout << "\*\*\*\* Enter last letters\n";**

**cin >> pok;**

**lpok = pok.size();// визначення довжини закінчення**

**// Пошук прізвищ із заданими літерами**

**cout << "\*\*\*\*\* Search name\n";**

**for (i = 0; i < n; i++)**

**{ lspis = spis[i].size(); // довжина прізвища**

**/\* spis[i].substr(0,lp) - це вибираються початкові літери прізвища**

**spis[i].substr(lspis-lp,lp) це вибираються кінцеві літери прізвища \*/**

**if (spis[i].substr(0,lp) ==p && spis[i].substr(lspis-lpok,lpok) == pok )**

**cout<<spis [i]<<endl; }**

**//-------------- Пошук прізвища з найменшою кількістю літер**

**minfam = spis[0].size(); /\* початкове значення з першого прізвища\*/**

**for (i = 1; i < n; i++)**

**if (spis[i].size() < minfam)**

**{ minfam = spis[i].size();**

**k=i; }**

**//-------------- Виведення результатів пошуку**

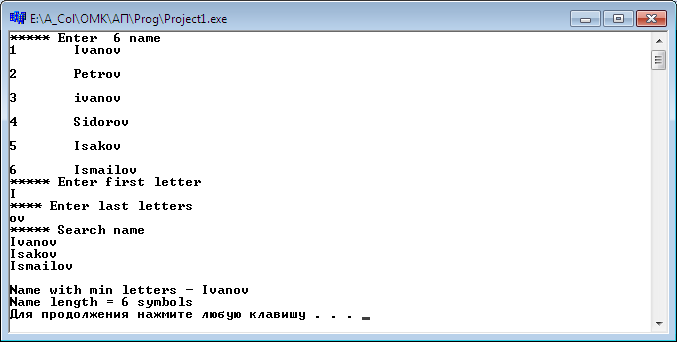
**cout<<"\nName with min letters - "<<spis [k]<<" \n";**

**cout << "Name length = " << spis [k].size() << " symbols \n";**

**system("pause");**

**return 0;**

**}**

Результати обчислень:  


У програмі для зберігання перших літер використовується змінна з ім’ям **р**, а для закінчення — змінна **роk**. Визначення перших літер та закінчення прізвища здійснює функція **size**. У цілому в операторі **if**… визначається як перші літери, так і закінчення прізвища. Змінна **minfam** існує для знаходження прізвища з найменшою кількістю літер згідно з алгоритмом визначення мінімального елемента масиву.

**Приклад 3.**

**#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**

