**Практична робота №7. Розгляд прикладів програм з символьними та рядковими величинами, масивами та структурами**

**Мета:** навчитися складати програми з символьними та рядковими величинами, масивами та структурами**.**

**Завдання до практичної роботи надається в кінці теоретичної частини.**

**Строк виконання 26.04.2020**

**ТЕОРЕТИЧНА ЧАСТИНА**

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.

***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

***Функція перетворення рядка типу string у рядок типу char:***

* **str.c\_str()** — перетворює рядок типу **string** у рядок типу **char.**

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

***Приклад 1.*** Увести список прізвищ і відсортувати його за алфавітом

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main()**

**{**

**const int n=5;**

**string spis[n];**

**int i, k;**

// Введення прізвищ

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

**{ cout <<"\*\*\* Enter "<<(i+1)<<" name\n";**

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

// Сортування прізвищ

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

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

**if (spis[i] > spis[i+1]) spis[i].swap(spis[i+1]);**

**//** Виведення результатів

**cout<<"\n\*\*\*\*\* Rezult spisok\n";**

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

**cout << (i+1) << ' ' << spis[i] << endl;**

**return 0;**

**}**

Результати виконання:  
\*\*\*\*\* Enter 1 name   
Котин С. Б.  
\*\*\*\*\* Enter 2 name   
Бобров М. Т.  
\*\*\*\*\* Enter 3 name   
Авдеев P. Л.  
\*\*\*\*\* Enter 4 name   
Попов Г. Д.  
\*\*\*\*\* Enter 5 name   
Елкин П. P.  
\*\*\*\*\* Rezult spisok  
1   Авдеев P. Л.  
2   Бобров M. Т.  
3   Елкин П. Р.  
4   Котин С. Б.  
5   Попов Г. Д.

Для порівняння прізвищ у програмі використовується звичайна операція «>», а для взаємозаміни — функція **swap()** (

***Приклад 2.*** 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

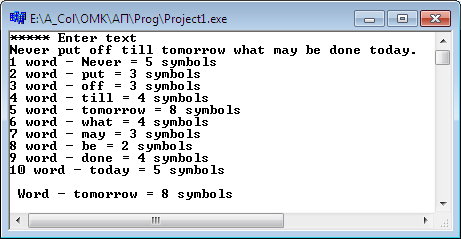
}

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

return 0;

}

Результат:



**Приклад 3.** 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

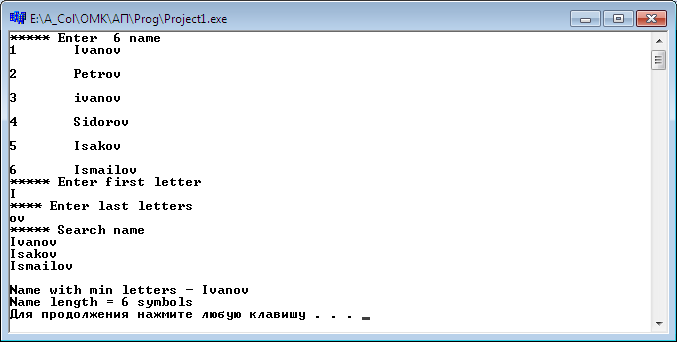
cout<<"\nName with min letters - "<<spis [k]<<" \n";

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.

**Контрольні запитання для письмових відповідей та завдання**

1. Які оператори введення-виведення символьних масивів ви знаєте? В яких випадках вони використовуються?
2. Як визначається кінець рядка?
3. Надайте список функцій перевірки рядків, які розглянуті на двох останніх заняттях.
4. Надайте список функцій маніпулювання рядками, які розглянуті на двох останніх заняттях.
5. Розгляньте приклад 3. Чи є у Вас зауваження щодо пошуку прізвища з найменшою кількістю літер? Чи всі прізвища будуть виведені, якщо ні, то що, на Вашу думку потрібно зробити?
6. В C++Shell Скласти програму оброблення рядків типу string**.**
7. Ввести для двох рядків такі речення,

Better late than never.

It is never too late to learn.

1. Визначте довжину кожного рядка і виведіть на екран разом з відповідним текстом.
2. Напишіть, які функції потрібно було б застосувати, щоб
3. об’єднати два речення,
4. знайти у першому реченні позицію слова «than»,
5. поставити перед словом «than» слово «learn».