

Лабораторна робота № 9.

Тема: Рядки. Масиви символів в C++.

Мета: Навчитися працювати з символьною інформацією, використовувати особливості побудови символьних рядків і функції стандартної бібліотеки C++ для роботи з ними.

Зміст роботи.

Написати програму мовою C++ з використанням вкладених циклів:

Завдання 1

Написати програму, яка друкує ваше прізвище та ім'я використовуючи різні способи введення та виведення даних. Визначте кількість символів у ваших даних.

Завдання 2

Дано рядок, довжина якого не перевищує n символів.

1. Усі повторні входження букви «а» у рядок замінити на букву «о». Якщо букви «а» у рядку немає, то вивести про це інформацію.
2. Підрахувати кількість входжень слова «мама» у рядок і вивести номери перших позицій цих входжень. Якщо цього слова у рядку немає, то вивести про це інформацію. Наприклад, рядок «мамамамама» містить 4 входження шуканого слова.
3. Підрахувати кількість входжень символів «+», «*» у рядок і вивести номери їх позицій. Якщо таких символів у рядку немає, то вивести про це інформацію.
4. Замінити усі символи «с» у рядку на «а». Якщо букв «с» у рядку немає, то вивести про це інформацію.
5. Замінити перший символ «к» рядка на символ «с», а останній символ «с» на «к». Якщо така заміна неможлива, то вивести про це інформацію.
6. Відшукати слово «ріпка» у рядку та замінити його на слово «колобок». Вивести номер початкової позиції виконаної заміни. Якщо така заміна неможлива, то вивести про це інформацію.
7. Підрахувати кількість відкритих та закритих круглих дужок у рядку та вивести номери їх позицій. Якщо дужок у рядку немає, то вивести про це інформацію.
8. Підрахувати кількість пробілів у рядку та вивести номери їх позицій. Перетворити рядок, викинувши пробіли. Якщо пробілів у початковому рядку немає, то вивести про це інформацію.
9. Підрахувати кількість відкритих та закритих квадратних дужок у рядку та вивести номери їх позицій. З'ясувати, чи є баланс дужок у рядку: для кожної дужки, яка відкривається, справа має бути дужка, що закривається. Якщо дужок у рядку немає, то вивести про це інформацію.
10. Замінити у рядку всі входження символу «н» на «nn». Якщо така заміна неможлива, то вивести про це інформацію.
11. Замінити у рядку всі входження «шч» на «щ». Якщо така заміна неможлива, то вивести про це інформацію.
12. Вивести номери позицій входження букви «и» у рядок. Вилучити з рядка всі входження цієї букви, крім останнього. Замінити останнє входження «и» на «е». Якщо букви «и» у рядку немає, то вивести про це інформацію.
13. Підрахувати кількість входжень символів «!», «?» у рядок і вивести номери їх позицій. Перетворити рядок, викинувши ці символи. Якщо таких символів у рядку немає, то вивести про це інформацію.
14. Підрахувати кількість відкритих та закритих фігурних дужок у рядку та вивести номери їх позицій. Вилучити усі символи, які розташовані всередині дужок. (Вважати, що для кожної відкритої дужки існує відповідна закрита, і між ними інших дужок немає). Якщо дужок у рядку немає, то вивести про це інформацію.

Завдання 3.

1. Дано ціле число $N (> 0)$ і рядок S . Перетворити рядок S у рядок довжиною N у такий спосіб: якщо довжина рядка S більше N , то відкинути перші символи, якщо довжина рядка S менше N , то в її початок додати символи «.» (крапка).
2. Дано цілі позитивні числа $N1$ й $N2$ і рядки $S1$ й $S2$. Одержати із цих рядків новий рядок, що містить перші $N1$ символів рядка $S1$ й останні $N2$ символів рядка $S2$ (у зазначеному порядку).
3. Дано символ C і рядок S . Подвоїти кожне входження символу C у рядок S .
4. Дано символ C і рядки S , $S0$. Перед кожним входженням символу C у рядок S вставити рядок $S0$.
5. Дано символ C і рядок S , $S0$. Після кожного входження символу C у рядок S вставити рядок $S0$.
6. Дано рядки S і $S0$. Знайти кількість входжень рядка $S0$ у рядок S .
7. Дано рядки S і $S0$. Видалити з рядка S перший підрядок, який співпадає з $S0$. Якщо співпадаючих підрядків нема, то вивести рядок S без змін.
8. Дано рядки S і $S0$. Видалити з рядка S останній підрядок, який співпадає з $S0$. Якщо співпадаючих підрядків нема, то вивести рядок S без змін.
9. Дано рядки S і $S0$. Видалити з рядка S всі підрядки, що співпадають з $S0$. Якщо співпадаючих рядків нема, то вивести рядок S без змін.
10. Дано рядки S , $S1$ і $S2$. Замінити в рядку S перше входження рядка $S1$ на рядок $S2$.
11. Дано рядки S , $S1$ і $S2$. Замінити в рядку S останнє входження рядка $S1$ на рядок $S2$.
12. Дано рядки S , $S1$ і $S2$. Замінити в рядку S всі входження рядка $S1$ на рядок $S2$.
13. Дано рядок, що містить принаймні один символ пробілу. Вивести підрядок, розташований між першим і другим пробілом вихідного рядка. Якщо рядок містить тільки один пробіл, то вивести порожній рядок.
14. Дано рядок, що містить принаймні один символ пробілу. Вивести підрядок, розташований між першим й останнім пробілом вихідного рядка. Якщо рядок містить тільки один пробіл, то вивести порожній рядок.
15. Дано рядки S й $S0$. Перевірити, чи входить рядок $S0$ у рядок S . Якщо входить, то вивести True, якщо не входить, то вивести False.

Теоретичні відомості.

Представлення символьної інформації в C++.

Важливо пам'ятати що:

'A'

"A"

1 – символ.

2 – рядок який складається з одного символу.

Тип char (скорочення від charakter – літера) віднесено до цілих базових типів. У більшості реалізацій C і C++ займає один байт і містить ASCII-код символу.

Тип string. Для оголошення символьних рядків вбудованого типу немає, але можна підключити заголовковий файл `#include` із стандартної бібліотеки STL, в якому визначений тип `string`.

Цей тип визначений як клас, структуру якого визначає масив, складений із елементів типу `char`. Для класу `string` визначені декілька конструкторів, низка операцій і методів. Отже, об'єкт типу `string` можна оголосити і потім йому присвоїти значення, а можна одразу ініціалізувати значенням рядкового літерала або іншого рядка, наприклад:

```
string str1, str2("some info"), str4="words", str3(str2);  
str1="sentence";
```

Для змінних типу `string` визначені операції присвоєння `=`, виведення за допомогою `cout<>`, конкатенації `+`, лексикографічного порівняння `<`, `<=`, `>=`, `==`, `!=`.

Наприклад:

```
string name, surname, fullname;  
name="John"; //присвоєння  
cout << "What is your surname? " << surname;  
//введення fullname=name+" "+surname;  
//конкатенація  
//виведення:
```

```
cout << "Your full name is " << fullname<< " "
```

рядок можна ввести тільки до першого пробільного символу. Тобто із речення буде введене тільки перше слово, решта залишиться у потоці до наступного застосування `cin>>`.

Для введення рядка із декількох слів використовують функцію **getline()**. Вона приймає два параметри: **перший** — **cin**, **другий** — **змінна типу string**.
Наприклад:

```
string name s1;  
cout << "Enter your full name: ";  
s1=getline(cin, name);  
cout << "Your full name is " << name<<s1.size()<<s1<<' '<<s1.length()<<endl;
```

Робота з символьним рядком як з масивом. З рядком можна також працювати і як із звичайним символьним масивом (говорять: в стилі C): оголошують масив типу `char` відповідного розміру з врахуванням нульсимволу як останнього символу рядка. Тоді можна користуватися функціями C. Функції C при роботі з рядками дуже ефективні, але достатньо незручні і небезпечні. Вони вимагають глибшого розуміння внутрішніх механізмів реалізації і не мають розвинутих засобів відстеження помилок. Рядковий літерал – константа типу символьного рядка в стилі C – послідовність довільних символів в подвійних лапках, обов'язково містить останній `'\0'`. Приклади рядкових літералів: `"some information"`, `"words"`. Приклади оголошення і ініціалізації символьних рядків

безпосередньо як масивів:

```
char str1[20]; str2[] = "1234", str3[]={ 'w','o','r','d','\0' };
```

char *strp="next sentence"; І в стилі C, і в стилі C++ для роботи з вмістом рядка можна застосовувати індексну форму, як для звичайного масиву, наприклад:

```
int i;
```

```
for(i=0;str2[i]!='\0';i++)
```

```
str1[i]=str2[i];
```

```
str1[i]='\0';
```

Використання рядків типу string

Для підключення цього класу до програми слід записати директиву:

#include <string> (без розширення **.h**)

і підключити простір імен бібліотеки шаблонів у вигляді

using namespace std; .

Існує багато функцій для обробки рядків типу **string**, розглянемо деякі з них.

Функції визначення довжини рядка:

- **str.size();**
- **str.length();**
- **str.max_size(); .**

Функції додавання одного рядка або його частини до іншого рядка:

- **Str.append(st);** — додає рядок **st** до кінця рядка **str**;
- **Str.append(st,k,n);** — додає до рядка **str** **n** символів рядка **st**, починаючи з позиції **k**.

Функція включення рядка в рядок:

- **str.insert(k,st)** — вставляє в рядок **str** з позиції **k** рядок **st**;
- **str.insert(k1,st,k2,n)** — вставляє в рядок **str** з позиції **k1** **n** символів рядка **st**, починаючи з позиції **k2** в рядку **st**.

Функція вилучення символів із рядка:

- **str.remove(k,n)** — вилучає **n** символів з рядка **str**, починаючи з позиції **k**.

Функція заміни частини рядка або усього рядка:

- **str.replace(st)** — заміняє рядок **str** на **st**;
- **str.replace(k,n,st)** — заміняє в рядку **str** **n** символів, починаючи з позиції **k** рядка **st**;
- **str.replace(k1,n1,st,k2,n2)** — заміняє в рядку **str** **n1** символів з позиції **k1** частиною **n2**символів рядка **st**, починаючи з позиції **k2**.

Функція обміну змістом двох рядків:

- **str.swap(st)** — обмінює зміст рядків **str** та **st**.

Функція виділення частини рядка:

- **str.substr(k,n)** — повертає частину рядка **str** в **n** символів, починаючи з позиції **k**.

Функція пошуку позиції входження підрядка в рядок:

- **str.find(st,k)** — шукає зліва граничну позицію входження рядка **st** в рядок **str**, починаючи з **k**-ої позиції рядка **str**;
- **str.rfind(st,k)** — шукає справа граничну позицію входження рядка **st** в рядок **str**, починаючи з **k**-ої позиції рядка **str**.

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**.

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.

Функція перетворення рядка типу string у рядок типу char:

- **str.c_str()** — перетворює рядок типу **string** у рядок типу **char**.

Таблиця 1 – Функції для роботи з рядками і символами

Функція	Пояснення
StrLen (ім'я_рядка)	визначає довжину вказаного рядка, без урахування нуль-символу
Копіювання рядків	
strcpy (S1, S2)	виконує побайтне копіювання символів з рядка s2 в рядок s1
strncpy (S1, S2, N)	виконує побайтне копіювання n символів з рядка s2 в рядок s1 . повертає значення s1
Конкатенація строк	
strcat (S1, S2)	об'єднує рядок s2 з рядком s1 . Результат зберігається в s1

Таблиця 1 – Функції для роботи з рядками і символами

Функція	Пояснення
strncat(S1, S2, N)	об'єднує n символів рядка s2 з рядком s1 . Результат зберігається в s1
Порівняння рядків	
strcmp(S1, S2)	порівнює рядок s1 з рядком s2 і повертає результат типу int : 0 - якщо рядки еквівалентні, >0 - якщо $s1 < s2$, <0 – якщо $s1 > s2$ З урахуванням регістру
strncmp(S1, S2, N)	порівнює n символів рядка s1 з рядком s2 і повертає результат типу int : 0 - якщо рядки еквівалентні, >0 - якщо $s1 < s2$, <0 – якщо $s1 > s2$ З урахуванням регістру
stricmp(S1, S2)	порівнює рядок s1 з рядком s2 і повертає результат типу int : 0 - якщо рядки еквівалентні, >0 - якщо $s1 < s2$, <0 – якщо $s1 > s2$ без урахування регістру
strnicmp(S1, S2, N)	порівнює n символів рядка s1 з рядком s2 і повертає результат типу int : 0 - якщо рядки еквівалентні, >0 - якщо $s1 < s2$, <0 – якщо $s1 > s2$ без урахування регістру
Обробка символів	
isalnum(c)	повертає значення true , якщо c є буквою або цифрою, і false в інших випадках
ISALPHA(c)	повертає значення true , якщо c є буквою, і false в інших випадках
isdigit(c)	повертає значення true , якщо c є цифрою, і false в інших випадках
ISLOWER(c)	повертає значення true , якщо c є буквою нижнього регістру, і false в інших випадках
ISUPPER(c)	повертає значення true , якщо c є буквою верхнього регістру, і false в інших випадках
isspace(c)	повертає значення true , якщо c є пробілом, і false в інших випадках
ToUpper(c)	якщо символ c , є символом нижнього регістра, то функція повертає перетворений символ c у верхньому регістрі, інакше символ повертається без змін.
Функції пошуку	
strchr(s,c)	пошук першого входження символу c в рядку s . У разі вдалого пошуку повертає покажчик на місце першого входження символу c . Якщо символ не знайдено, то повертається нуль.

Таблиця 1 – Функції для роботи з рядками і символами

Функція	Пояснення
strcspn(S1, S2)	визначає довжину початкового сегмента рядка s1 , який містить символи, що не входять в рядок s2
strspn(S1, S2)	повертає довжину початкового сегмента рядка s1 , який містить тільки ті символи, які входять в рядок s2
strprbk(S1, S2)	Повертає покажчик першого входження будь-якого символу рядка s2 в рядку s1
Функції перетворення	
atof(S1)	перетворить рядок s1 в тип double
atoi(S1)	перетворить рядок s1 в тип int
atol(S1)	перетворить рядок s1 в тип long int
Функції стандартної бібліотеки вводу / виводу <stdio>	
GetChar(з)	зчитує символ c зі стандартного потоку введення, повертає символ у форматі int
gets(s)	зчитує потік символів зі стандартного пристрою введення в рядок s доти, поки не буде натиснута клавіша ENTER

Заголовковий файл <ctype.h>, <cctype> містить групу функцій, призначених для класифікації символів типу char та їх перетворення.

Прототип	Призначення
<code>int isalnum(int c);</code>	символ <code>c</code> є ASCII-кодом латинської букви або цифри ('a'-'z', 'A'-'Z', '0'-'9')
<code>int isalpha(int c);</code>	символ <code>c</code> є ASCII-кодом латинської букви ('a'-'z', 'A'-'Z')
<code>int isascii(int c);</code>	символ <code>c</code> є ASCII-кодом (0-127)
<code>int isdigit(int c);</code>	символ <code>c</code> є ASCII-кодом десяткової цифри ('0'-'9')
<code>int iscntrl(int c);</code>	символ <code>c</code> є ASCII-кодом керуючого знаку (0x7f або 0x00-0x1f)
<code>int islower(int c);</code>	символ <code>c</code> є ASCII-кодом малої букви ('a'-'z')
<code>int isprint(int c);</code> <code>int isgraph(int c);</code>	символ <code>c</code> є ASCII-кодом видимого знаку (0x20-0x7e), (крім пропуску 0x20 для <code>isprint</code>)
<code>int ispunct(int c);</code>	символ <code>c</code> є ASCII-кодом розділового знаку
<code>int isspace(int c);</code>	символ <code>c</code> є ASCII-кодом пропуску
<code>int isupper(int c);</code>	символ <code>c</code> є ASCII-кодом великої букви ('A'-'Z')
<code>int isxdigit(int c);</code>	символ <code>c</code> є ASCII-кодом шістнадцяткової цифри ('0'-'9', 'a'-'f', 'A'-'F')
<code>int toascii(int c);</code>	перетворює цілі числа в ASCII-коди
<code>int tolower(int c);</code>	перетворює великі латинські букви у маленькі
<code>int toupper(int c);</code>	перетворює маленькі латинські букви у великі

Приклад. Дано текстовий масив `A(5)`. Знайти і надрукувати найменшу довжину. Вивести на друк елементи з найменшою довжиною та їх індекси (порядкові номери).

Розв'язання:

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    const int n = 5;
    string A[n];
    cout << "Введіть текстовий масив" << endl;
    for (int i=0; i<n; i++)
    {
        cout<< "A[" << i << "]=";
        cin >> A[i];
    }
}
```

```

int minLen = A[0].length();
// визначаємо найменшу довжину
for (int i=0; i<n; i++)
{
    int len=A[i].length();
    if(len < minLen)
        minLen = len;
}
cout << "Найменша довжина " << minLen << endl;
// Роздруковуємо елементи з найменшою довжиною і їх індекси
for (int i=0; i<n; i++)
{
    int len=A[i].length();
    if( len == minLen)
        cout<<"індекс="<< i <<" елемент= " << A[i] << endl;
}

return 0;
}

```

Контрольні запитання.

1. Як описуються символьні змінні?
2. Як зберігаються текстові рядки в мові C++?
3. Як записуються символьні константи?
4. Як записуються керуючі символи?
5. Які засоби використовуються для введення/виведення символів?
6. Охарактеризуйте основні функції обробки символьних даних.
7. Як записуються рядкові константи?
8. Як обчислюється довжина рядка?
9. Перелічіть деякі функції копіювання рядків.
10. Які функції використовуються для зчеплення рядків?
11. Які функції використовуються для пошуку в рядках?
12. Як здійснюється порівняння рядків?
13. Як можна здійснити перетворення рядка до числа?
14. Як здійснюється перетворення числа до рядкового подання?