### . Різновиди констант

У цьому підрозділі наведено мінімальні відомості про вигляд констант мови С++ і засоби їх виведення.

#### 2.5.1. Символьні константи

Символьна константа – це символ в апострофах **('A', '1', '.')** або два символи в апострофах, першим з яких є \ (зворотна скісна, або *backslash*). Наприклад, константи **'\'', '\"', '\\'** позначають символи, що називаються "апостроф", "лапки" та "зворотна скісна". Є кілька констант, наприклад **'\n'** або **'\t',** що містять малу латинську літеру (**керувальні символи**). Вони використовуються в спеціальний спосіб під час виведення на зовнішні носії даних. Зокрема, **'\n'** задає перехід на новий рядок екрана.

Символьні константи зображують значення типу **char**, які займають один байт. Цей байт як зображення числа без знака дає **код символу** (число від 0 до 255), а зі знаком – від -128 до 127. Відповідність між числами від 0 до 127 і символами зафіксовано в *Американському стандартному коді для обміну інформацією* (так звана *таблиця ASCII*), решті кодів можуть відповідати різні набори символів. Наприклад, константа ' ' (пробіл) має код 32, а '**\0**' – код 0. Окремими властивостями таблиці ASCII є такі:

* символам '**0**'**,** '**1**'**, …,** '**9**' відповідають послідовні коди від 48 до 57;
* символам **'A', 'B',** …**, 'Z'** – від 65 до 90; – символам **'a', 'b',** …, **'z'** – від 97 до 122.

Інструкція **cout<<'Z';** виводить значення константи **'Z'** – на екрані з'являється **Z**. Аналогічно можна вивести інші символьні константи. Для того, щоб наступне повідомлення виводилося з нового рядка, можна скористатися інструкцією **cout<<endl;** або **cout<<'\n'; .**

За інструкцією вигляду **cout<<*константа*;** обробляється не константа, а зображене нею значення. Послідовність символів, утворена за значенням, може відрізнятися від константи.

#### Вправи

2.2. Що буде виведено за програмою?

**#include <iostream> using namespace std; int main() { cout<<'1'; cout<<'\n'; cout<<'Y'; cout<<'e'; cout<<'s'; cout<<endl; system("pause"); return 0;**

**}**

2.3. Написати програму, що друкує символи **', ", \.**

##### 2.5.2. Рядкові константи

Рядкові константи позначають послідовності символів і записуються в лапках, наприклад **"You are welcome!".** Символи **'**, **\**, **"** у рядковій константі записують як **\'**, **\\**, **\"**, відповідно. До значення, заданого рядковою константою, у пам'яті додається символ **'\0'**, який позначає кінець послідовності символів. Окрім того, якщо в константі присутній символ **'\0'**, то значенням, яке вона задає, є послідовність символів перед **'\0'**. Наприклад, константі **"\"A'BC\"\0zz"** відповідає послідовність байтів із символами

**" A ' B C " '\0' z z '\0'**

Значення утворене першими шістьма символами **"A'BC".** Саме воно виводиться за інструкцією **cout<<"\"A'BC\"\0zz";** .

За допомогою рядкових констант перепишемо програму зі вправи 2.2.

**#include <iostream> using namespace std; int main() { cout<<"1\nYes\n"; system("pause"); return 0;**

**}**

**prog003.cpp**

#### Вправи

2.4. Написати програму, що виводить ім'я дискового каталогу, в якому встановлено систему програмування C++.

2.5. Поясніть різницю між значеннями **"a"** та **'a'**.

##### 2.5.3. Цілі константи

**Цілі константи** позначають цілі числа й мають десяткову, вісімкову й шістнадцяткову форму запису (про системи числення див. додаток А). У мові C++ цілі константи невід'ємні. **Десяткова константа** – це, як і в математиці, послідовність десяткових цифр, що не починається з **0**: **273**, **1024** тощо. Запис **шістнадцяткових констант** починається символами **0x** або **0X**. Наприклад, константа **0x11** позначає число 17, а **0xf** та **0xF** – число 15. Ціла константа, що починається з **0** й далі містить цифри від 0 до 7, є **вісімковою**. Наприклад, константи **0** та **010** позначають числа 0 та 8, відповідно.

Якщо в програмі спеціально не вказано інше, значення цілих констант незалежно від форми їх запису *виводяться в десятковому записі*. Наприклад, за інструкцією **cout<<11;** на екран виводиться **11,** а за **cout<<0x11; – 17**.

#### Вправи

2.6. Пояснити різницю між а) **0, '\0'** та **'0';** б) **123** та **"123"**.

2.7. Що виводиться на екран за такими двома програмами? Поясніть різницю між ними.

**#include <iostream> #include <iostream> using namespace std; using namespace std;**

**int main() { int main() {**

**cout<<13; cout<<'\n'; cout<<"13"; cout<<'\n'; cout<<013; cout<<'\n'; cout<<"013"; cout<<'\n'; cout<<0x13; cout<<endl; cout<<"0x13"; cout<<endl;**

**system("pause"); system("pause");**

**return 0; return 0;**

**} }**

##### 2.5.4. Дійсні константи

Розглянемо приклад. Число 12,34 можна позначити, наприклад, так: 123,410-1. У цьому записі є ціла частина **123**, дробова частина **,4** і десятковий порядок **–1.** Запису відповідає константа **123.4E-1,** в якій **123** – *ціла* частина, **.4** – *дробова*, а **E–1** – *порядок*.

Це саме число можна зобразити й іншими константами: **12.34**,

**1234.e-03**, **1234e-03**, **.1234E2**, **0.01234E+3** тощо. Отже, ціла частина – це непорожня послідовність цифр, дробова – послідовність цифр із *крапкою* на початку, порядок – латинська літера **E** або **e** з однією або кількома цифрами, можливо, зі знаком **+** або **-**.

Константа, в якій ціла частина містить одну цифру від

**1** до **9,** називається **нормалізованою**, наприклад

**1.234E+1, 9.81** або **1.0E2** (для числа 0 такою є **0.0**).

*Дійсні константи записуються тільки як десяткові*. Якщо константа має цілу частину, то за нею йде дробова частина й порядок (можливо, одне з них). При цьому в дробовій частині, якщо вона є, цифр може не бути, але крапка обов'язкова. Якщо цілої частини немає, то константа починається крапкою, після якої має бути хоча б одна цифра. Порядок необов'язковий. Приклади наведено вище. Перед константою може бути знак **–**; тоді вона задає від'ємне число: **–1.2345E1.**

Значення дійсних констант виводяться в різних формах (*форматах*). Значення, зображені дійсними константами, виводяться або з **фіксованою** (*fixed*) **крапкою**, або в **нормалізованій** (**науковій**– *scientific*) **формі**; наприклад, для числа 12,34 це відповідно **12.34** та **1.234e+001.** Від чого залежить формат виведення та як ним керувати, див у п. 3.3.2.