Базові складові мови C/C++

- Алфавіт (абетка)
- Лексеми
- Вирази
- Оператори

*У природній мові спілкування виділяють чотири основні елементи: **символ**, **слово**, **словосполучення** та **речення**. Подібні елементи існують і в алгоритмічній мові, тільки слова мають назву **лексеми**, словосполучення — **вирази**, а речення — **оператори**. Лексеми створюються із символів, вирази — із лексем та символів, оператори — з символів, виразів і лексем.*

Алфавіт

Алфавіт мови С++ включає:

- великі (**A-Z**) і малі (**a—z**) літери латинського алфавіту та символ підкреслення (**_**);
- арабські цифри від **0** до **9**;
- знаки арифметичних дій **+, -, *, /, %, ++, --**;
- знаки побітових операцій **<<, >>, &, |, ~, ^**;
- знаки відношень **<, <=, ==, !=, >, >=**;
- знаки логічних операцій **&&, ||, !**;
- розділові знаки **, ; : пропуск**;
- спеціальні знаки **., =, ->, ?, \, $, #, ', "**;
- символи дужок **(,), [,], {, }**.

Лексеми та ідентифікатори

- **Лексеми**, тобто базові елементи мови з певним самостійним значенням, складаються із символів алфавіту. До них відносять ідентифікатори, ключові слова, знаки операцій, константи, роздільники (дужки, крапка, кома, символи пропуску). Межі лексем визначаються іншими лексемами-роздільниками або знаками операцій.
- ***Ідентифікатором***, тобто ім'ям програмного об'єкта, називається будь-яка послідовність літер латинського алфавіту, цифр і символу підкреслення за умови, що першою стоїть літера або символ підкреслення, а не цифра.

Існує два різновиди ідентифікаторів:

- *стандартні*, наприклад, імена всіх вбудованих у мову функцій;
- *користувальницькі (створені користувачем-програмістом)*.

Примітка: в C/C++ при створені ідентифікаторів розрізняються великі та малі символи (на відміну від інших мов, наприклад Pascal, SQL,…), тобто наступні ідентифікатори абсолютно різні і мають право на існування в одному блоку:

min, mIn, Min, MIN.

Ключові слова (зарезервовані ідентифікатори)

- ***Ключовими*** (службовими) словами називають ряд зарезервованих ідентифікаторів, що вживаються для побудови конструкцій мови і мають фіксоване значення. За смисловим навантаженням службові слова поділяються на такі основні групи:
- специфікатори типів — **char, int, long, typedef, short, float, double, enum, struct, union, signed, unsigned, void;**
- кваліфікатори типів — **const і volatile;**
- класи пам’яті — **auto, extern, register, static;**
- для побудови операторів — **for, while, do, if, else, switch, case, continue, goto, break, return, default, sizeof.**

Основні ключові слова C++

asm	delete	goto	register	throw
auto	do	if	return	try
break	double	inline	short	typedef
case	else	int	signed	typename
catch	enum	long	sizeof	union
char	explicit	new	static	unsigned
class	extern	operator	struct	virtual
const	float	private	switch	void
continue	for	protected	template	volatile
default	friend	public	this	while

Структура простої програми мовою C/C++

- Підключення бібліотек (#include <им’я файлу>)
- Головна функція main. Наявність функції main обов'язкове , причому в одному екземплярі – це «місце входу» виконання програми.
- Тіло (блок {}) головної функції main.

```
#include <им’я файлу> // Підключення бібліотек
int main() // головна функція main
{ //початок тіла main
        оператори
        return <ціле значення>; // головна функція по стандарту повинна вертати ціле значення
} //кінець тіла main
```

Примітка: по стандарту існує тільки два вигляди головної функції main:

- *int main()*
- *int main(int argc, char *argv[])*

*Варіант **void main()**, який достатньо часто зустрічається в літературі, **не по стандарту** і **не підтримується** усіма компіляторами, наприклад, таких як **gcc**.*

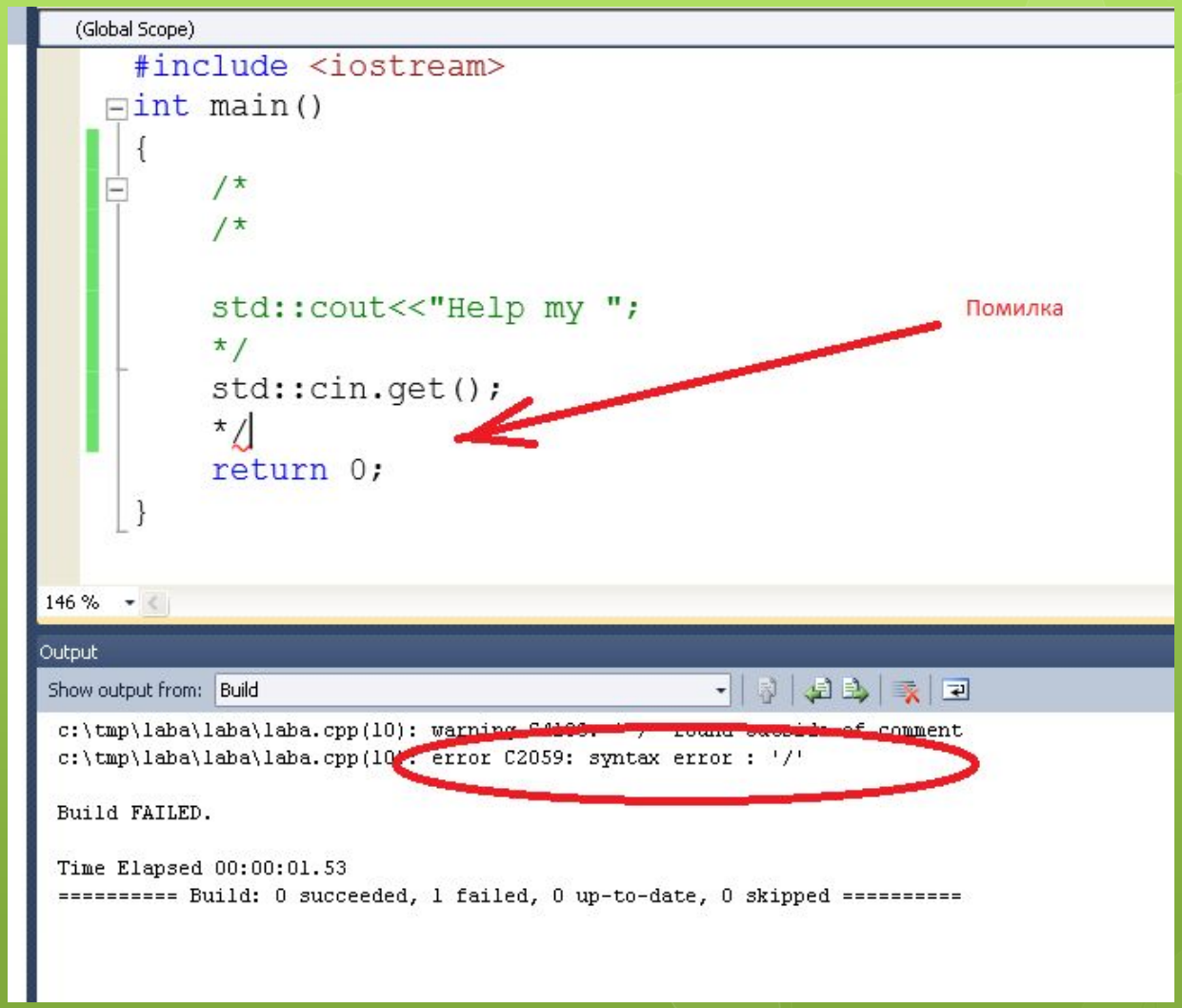
Приклад пустої програми, яка компілюється, но нічого не робить *(її доцільно застосовувати для перевірки працездатності компілятора)*:

```
int main()
{
     return 0;
}
```

Коментарі

- **Коментарі** необхідні для пояснень призначення тих чи інших частин програми і їх текст завжди ігнорується компілятором. Мова C/C++ використовує два різновиди коментарів:
- *// текст* — **однорядковий коментар**, який починається з двох символів «/» («коса риска») і закінчується символом переходу на новий рядок;
- */* текст */* — **багаторядковий коментар**, що розташовується між символами-дужками «/*» і «*/».
- Багаторядкові коментарі не можуть бути вкладеними один в один, а однорядкові коментарі можна вкладати в багаторядкові коментарі.

Наприклад, таке вкладене застосування коментаря приведе до наступної помилки:



Створення змінної

- **Змінна** — це іменована область пам’яті, у якій зберігаються дані визначеного типу. Змінна має ім’я, розмір та інші атрибути, такі як видимість, час існування тощо. Ім’я змінної служить для звертання до області пам’яті, у якій зберігається її значення. ***Перед використанням будь-яка змінна повинна бути описана***, при цьому для неї резервується деяка область пам’яті, розмір якої залежить від конкретного типу змінної. Під час виконання програми змінна може приймати різні значення.

Загальний вигляд опису змінних:

[клас пам’яті] [const] тип ім’я [ініціалізація];

static int a=5;//створення змінної/об’єкту ‘a’.

Спрощений синтаксис створення змінної:

тип ім’я;

Клас пам’яті

Клас пам’яті визначає час існування та область видимості програмного об’єкта, тобто змінної. Якщо клас пам’яті не зазначений явно, то він визначається компілятором (зазвичай як auto), виходячи, з контексту оголошення.

- **auto** — автоматична змінна, для якої пам’ять виділяється у стеку і за необхідності ініціюється кожного разу при виконанні оператора, що містить її визначення. Звільнення пам’яті відбувається при виході з блока, де описана змінна. Час її існування — з моменту опису до кінця виконання блока. Для глобальних змінних цей специфікатор не використовується, а для локальних він приймається за замовчуванням, тому задавати його явно великого сенсу немає;

  (**розпочинаючи зі стандарту c++11 значення auto змінено**)

- **extern** означає, що змінна визначена в іншому місці програми (в іншому файлі або далі по тексту) і використовується для створення змінних, доступних в усіх модулях програми, де вони оголошені. При ініціюванні змінної у тому ж операторі, специфікатор **extern** ігнорується;

- **static** — статична змінна, що має постійний час існування. Вона ініціюється один раз при першому виконанні оператору, що містить визначення змінної. Залежно від розташування оператора, описані статичні змінні можуть бути глобальними і локальними. Глобальні статичні змінні видимі тільки у тому модулі, в якому вони описані;

- **register** — аналогічний до специфікатора **auto**, але пам’ять виділяється по можливості в регістрах процесора і за відсутності такої можливості у компілятора змінні обробляються як **auto**.

Створення константи

- модифікатор **const** вказує, що змінна не може змінювати своє значення, у цьому випадку її називають **типізованою (іменованою) константою** або просто **константою**;
- ініціалізація для констант обов’язкова.

Наприклад:

```
const int a = 5;
int const b = 7;
const double pi = 3.14;
```

Глобальні/локальні змінні

- ***Локальна змінна*** визначена всередині блока. Область її дії обмежена початком опису змінної та кінцем блока, включаючи усі вкладені блоки. Час “життя” з моменту опису до кінця блоку в якому вона об’явлена.

```
{
    int a;//локальна змінна
}
```

- Блок – це частина коду розташована у фігурних дужках {}.
- ***Глобальна змінна*** - це змінна, визначена поза будь-яким блоком. Областю її дії вважається файл, у якому вона визначена від початку опису до його кінця. Час “життя” з моменту опису до кінця існування програми.

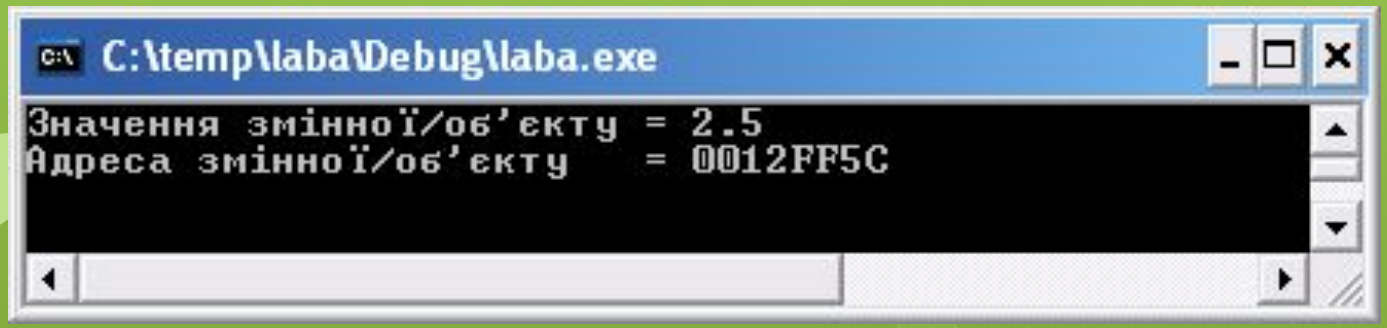
```
int a;//глобальна змінна
int main()
{ ……}
```

Приклади створення змінних

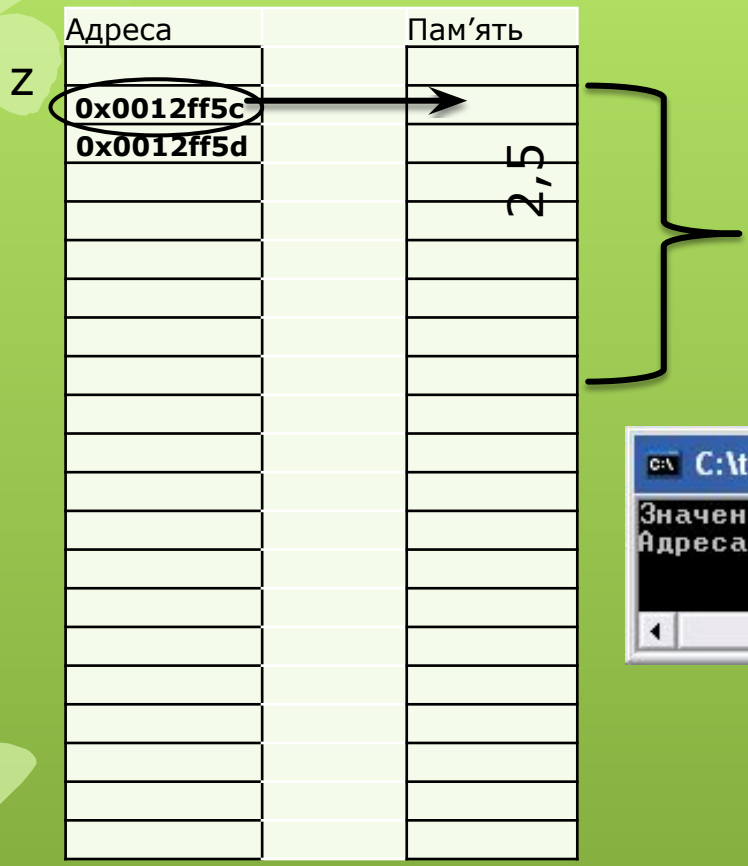
```
int d;                    //1 — глобальна змінна d
int main()
{
      int b;          //2 — локальна змінна b
      extern int y;       //3 — змінна, визначена у іншому місці
                  //(файлі) програми
      static int s;   //4 — локальна статична змінна s
      d = 1;          //5 — присвоювання  значення глобальній змінній
      int d;          //6 — створення локальної змінної d
      d = 10;               //7 — присвоювання  значення локальній змінній
      ::d = 3;              //8 — присвоювання  значення глобальній змінній
      return 0;
}
int y = 4;                // 9 — визначення і ініціалізація змінної y
```

Представлення змінної/об’єкту в пам’яті

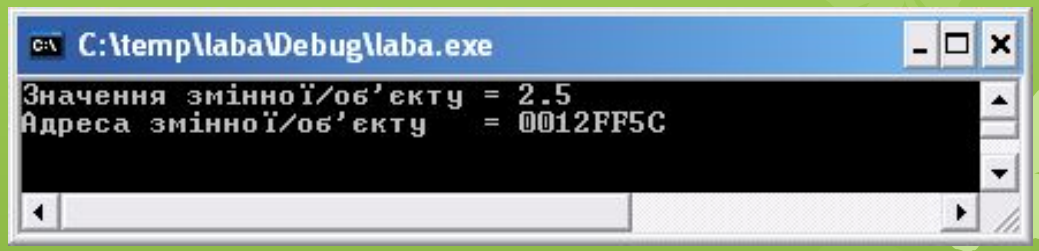
```
#include <iostream>
int main()
{
    setlocale(LC_ALL, "Ukr");
    double x= 2.5;
    std::cout<<"Значення змінної/об'єкту = "<<x<<std::endl;
    std::cout<<"Адреса змінної/об'єкту   = "<<&x<<std::endl;
    std::cin.get();
    return 0;
}
```



Представлення об’єкту в пам’яті

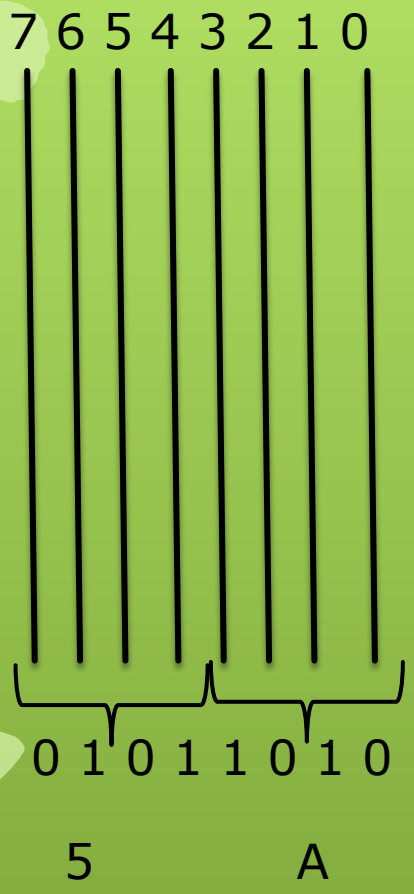


double z =2.5



- Машинне слово - машинно-залежна і платформозалежна величина, яка вимірюється в бітах або байтах (тритій або трайтен), рівна розрядності регістрів процесора і/або розрядності шини даних (зазвичай деяка ступінь двійки)
- Байт (англ. Byte) - одиниця зберігання і обробки цифрової інформації; сукупність бітів, що обробляється комп'ютером одномоментно. В сучасних обчислювальних системах байт складається з восьми бітів і, відповідно, може приймати одне з 256 (28) різних значень (станів, кодів). Однак в історії комп'ютерної техніки існували рішення з іншими розмірами байту (наприклад, 6, 32 або 36 бітів), тому іноді в комп'ютерних стандартах і офіційних документах для однозначного позначення групи з 8 бітів використовується термін «октет» (лат. Octet). У більшості обчислювальних архітектур байт - це мінімальний незалежно адресуємий набір даних.
- Біт - одиниця виміру кількості інформації. Може приймати тільки два взаємовиключних значення: «так» або «ні», «1» або «0», «включено» або «вимкнено».

Структура байту (8-ми бітного)



bin	dec	hex	oct
0 0 0 0	0	0	0
0 0 0 1	1	1	1
0 0 1 0	2	2	2
0 0 1 1	3	3	3
0 1 0 0	4	4	4
0 1 0 1	5	5	5
0 1 1 0	6	6	6
0 1 1 1	7	7	7
1 0 0 0	8	8	10
1 0 0 1	9	9	11
1 0 1 0	10	A	12
1 0 1 1	11	B	13
1 1 0 0	12	C	14
1 1 0 1	13	D	15
1 1 1 0	14	E	16
1 1 1 1	15	F	17

```
#include <iostream>
int main()
{
    int a = 90;
    std::cout<<std::dec<<a<<std::endl;
    std::cout<<std::hex<<a<<std::endl;
    std::cout<<std::oct<<a<<std::endl;
    std::cin.get();
    return 0;
}
```

Відомі трітові комп'ютери

- У 1970 році Брусенцовим була створена нова машина «Сетунь-70».
- У 1973 році в США побудований експериментальний трійчастий комп'ютер TERNAC, основними завданнями якого були перевірка реалізації недвійковий структур на двійковому комп'ютері і порівняння техніко-економічних характеристик з двійковими аналогами.
- У 2008 році побудована цифрова 3-х трітова комп'ютерна система TCA2.
- Що стосується нинішнього часу, достовірних фактів використання або розробки сучасних трійчастих ЕОМ не виявлено - є певний інтерес з боку компаній, наприклад, IBM, але відсутня активна участь або освітлення цієї участі в пресі. Хоча, відомий американський учений, професор Дональд Кнут вважає, що тризначна логіка елегантніша і ефективніша двійкової і, можливо, в майбутньому, розробки будуть продовжені.

Основні типи даних

- **int** (цілий);
- **char** (символьний);
- **bool** (логічний);
- **float** (дійсний);
- **double** (дійсний з подвійною точністю);
- **void** (порожній, не має значення).

Типи **int, char, bool** називають ***цілими***, а типи **float** та **double** — ***дійсними з плаваючою крапкою***. Код, що формує компілятор для обробки цілих величин, відрізняється від коду для величин з плаваючою крапкою.

Для уточнення внутрішнього подання та діапазону значень стандартних типів **мова C++ використовує чотири специфікатори типу**:

- **short** (короткий);
- **long** (довгий);
- **signed** (знаковий);
- **unsigned** (беззнаковий).

Зауваження: символьний (char) та логічний (bool) тип також відносять до цілого.

Зауваження: Нові версії компіляторів мають більш розширені типи, наприклад, long long, wchar_t.

Основні типи даних

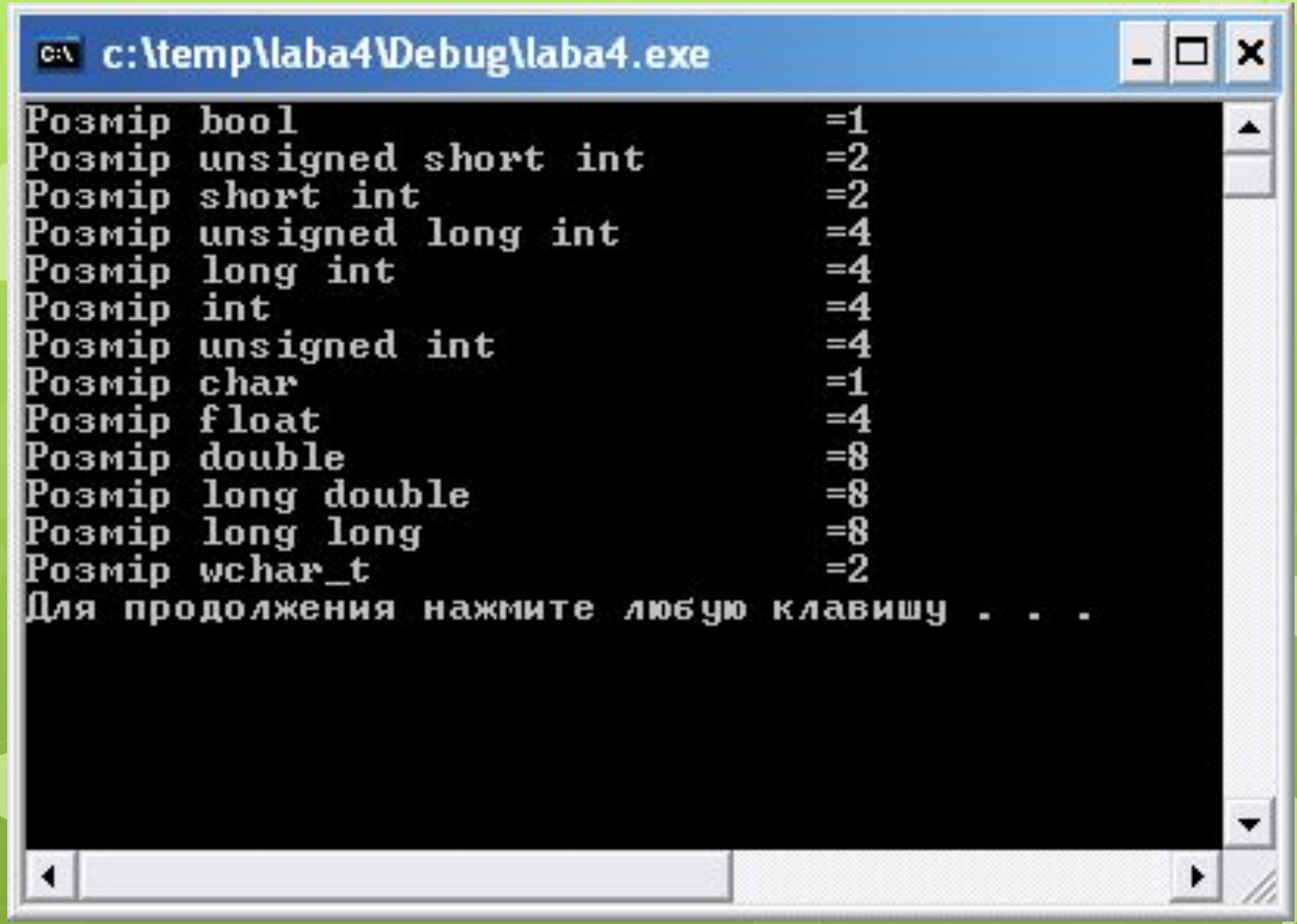
- ***Цілі змінні*** (типу **int, long, short**) необхідні для збереження цілих значень і можуть бути знаковими і беззнаковими. Знакові змінні застосовують для подання як додатних, так і від'ємних чисел, при цьому один біт (найстарший) виділяється під знак. Для оголошення беззнакової змінної, тобто змінної, що приймає тільки додатні значення, необхідно використовувати ключове слово **unsigned**. За замовчуванням будь-який цілий тип вважається знаковим, і тому немає потреби у використанні ключового слова **signed**.
- ***Символьний тип*** даних **char** застосовується у випадку, коли змінна містить інформацію про код ASCII або для побудови таких більш складних конструкцій, як рядки, символьні масиви тощо. Дані типу **char** також можуть бути знаковими і беззнаковими.
- ***Змінна типу*** **bool** займає 1 байт і використовується, насамперед, у логічних операціях, тому що може приймати значення **0** (**false** — «неправда») або відмінне від нуля (**true** — «істина»). У випадку перетворення до цілого типу **true** має значення 1.
- Стандарт C++ визначає три типи даних для збереження ***дійсних значень змінних***: **float, double** та **long double** (типи ***з плаваючою крапкою***). Тип **float**, як правило, використовують для збереження не дуже великих дробових чисел.
- ***Змінна типу*** **void** не має значення, оскільки множина значень цього типу порожня. Такі змінні необхідні для узгодження синтаксису. Тип **void** використовується для визначення функцій, що не повертають значення, для вказівки порожнього списку аргументів функції, а також як базовий тип для покажчиків і в операції приведення типів. Наприклад, якщо немає потреби у використанні поверненого значення функції, перед ім'ям функції ставиться тип **void**.

Основні типи даних

Тип	Розмір, байт	Значення
bool	1	true або false
unsigned short int	2	від 0 до 65 535
short int	2	від -32 768 до 32 767
unsigned long int	4	від 0 до 4 294 967 295
long int	4	від -2 147 483 648 до 2 147 483 647
int (16 розрядів)	2	від -32 768 до 32 767
int (32 розряди)	4	від -2 147 483 648 до 2 147 483 647
unsigned int (16 розрядів)	2	від 0 до 65 535
unsigned int (32 розряди)	4	від 0 до 4 294 967 295
char	1	від 0 до 256
float	4	від 1.2e-38 до 3.4e38
double	8	від 2.2e-308 до 1.8e308
long double	8/10	від 3.4e-4932 до 3.4e+4932

Програми яка друкує розмір типу в байтах

```
#include <iostream>
using namespace std;
int main()
{
	setlocale(LC_ALL, "Ukr");
	cout<<"Розмір bool \t\t\t="<<sizeof( bool )<<endl;
	cout<<"Розмір unsigned short int \t="<<sizeof(unsigned short int)<<endl;
	cout<<"Розмір short int \t\t="<<sizeof(short int)<<endl;
	cout<<"Розмір unsigned long int \t="<<sizeof(unsigned long int)<<endl;
	cout<<"Розмір long int \t\t="<<sizeof(long int)<<endl;
	cout<<"Розмір int \t\t\t="<<sizeof(int )<<endl;
	cout<<"Розмір unsigned int \t\t="<<sizeof(unsigned int)<<endl;
	cout<<"Розмір char \t\t\t="<<sizeof(char)<<endl;
	cout<<"Розмір float \t\t\t="<<sizeof(float)<<endl;
	cout<<"Розмір double \t\t\t="<<sizeof(double)<<endl;
	cout<<"Розмір long double \t\t="<<sizeof(long double)<<endl;
	cout<<"Розмір long long \t\t="<<sizeof(long long)<<endl;
	cout<<"Розмір wchar_t \t\t\t="<<sizeof(wchar_t)<<endl;
	system("pause");
	return 0;
}
```


c:\temp\laba4\Debug\laba4.exe
Розмір bool =1
Розмір unsigned short int =2
Розмір short int =2
Розмір unsigned long int =4
Розмір long int =4
Розмір int =4
Розмір unsigned int =4
Розмір char =1
Розмір float =4
Розмір double =8
Розмір long double =8
Розмір long long =8
Розмір wchar_t =2
Для продолжения нажмите любую клавишу . . .