Лабораторна робота № 4 Програмування найпростіших обчислювальних алгоритмів лінійної структури

Мета роботи: навчитися складати й програмувати мовою С найпростіші обчислювальні алгоритми лінійної структури, а також визначати константи, використовувати функції стандартної математичної бібліотеки math.h і формати функцій printf і scanf при введенні-виведенні даних.

Завдання: дано алгоритм, відповідно до якого треба послідовно виконати такі дії:

— вивести на екран вигляд заданих функцій f_1, f_2, f_3;

— ввести значення параметра a і змінної x;

— трьом змінним y, z, t дійсного типу послідовно присвоїти значення трьох заданих функцій (при цьому **константи для першої функції** f_1 **визначити в декларативній частині**, **функцію** f_2 **записати з мінімумом операцій, функцію** f_3 **запрограмувати без оптимізації**), **ОДЗ не перевіряти**;

— вивести обчислені значення (y=значення_y) на екран у форматі з фіксованою точкою;

— виконати переприсвоєння значень змінних ($y \leftarrow z$, $z \leftarrow t$, $t \leftarrow y$);

— знову вивести на екран значення змінних (y=значення_y) у форматі з фіксованою точкою.

Намалювати блок-схему алгоритму і запрограмувати його мовою С. Текст програми подати в структурованому вигляді. Коментарі в програмі обов'язкові (17-25%; не забувайте в коментарях писати **прізвище, групу, варіант, назву роботи**). За алгоритмом провести розрахунки не менш ніж з трьома різними наборами значень параметра a і змінної x (значення a і x підбирати так, щоб вони задовольняли ОДЗ). Правильність обчислень перевірити в Excel (при захисті роботи показувати результат розрахунку). У звіті формули набирати лише в Microsoft Equation.

Варіанти завдань (номер варіанту відповідає номеру студента за списком)

1. $\frac{\sqrt{x-2}}{3-ax}+|\cos x-4|$, $\frac{1}{4}x^3+\frac{3}{10}x^2+7e^{1-tgx}$, $\frac{7}{8}x^{-0,12}$

2. $\frac{x-a}{1-7\sqrt{x}}-\log_5|x-1|$, $\frac{3}{4}x^2-\frac{7}{10}x+2e^{-tgx}$, $\frac{3}{5}x^{-5,3}$

3. $\frac{3x-2a}{\sqrt{x}+5}-\ln|x+2|$, $\frac{5}{2}x^2-\frac{7}{10}x-e^{1-ctgx}$, $\frac{4}{5}x^{-2,1}$

4. $\sqrt{\frac{x-a}{3x+1}}+|\sin x-2|$, $\frac{1}{4}x+\frac{7}{5}x^3-2e^{x-3}$, $\frac{7}{10}x^{4,2}$

5. $\sqrt{\frac{\ln x+1}{5x+a}}+|x+5tgx|$, $\frac{1}{8}x-\frac{3}{10}x^2-e^{ctgx+1}$, $\frac{5}{8}x^{-4,1}$

6. $\frac{7x-a}{3+\log_2 x}+\sqrt{|x+3|}$, $\frac{1}{4}x^2-\frac{4}{10}x^3-e^{3x+5}$, $\frac{4}{5}x^{-0,3}$

7. $\sqrt{\frac{5x-a}{x^2-1}}+\cos|x-1|$, $\frac{5}{4}x^2-\frac{4}{20}x-2e^{\sin x}$, $\frac{4}{5}x^{5,4}$

8. $\frac{\sqrt{\ln x+3}}{3x+a-4}+ctg|x-4|$, $\frac{7}{4}x^3-\frac{4}{5}x+xe^{-3tgx}$, $\frac{7}{8}x^{0,7}$

9. $\frac{\sqrt{x+3}}{2a-x}+\log_3|\cos x+3|$, $\frac{8}{10}x+\frac{7}{4}x^3-e^{1+x}$, $\frac{3}{8}x^{-4,4}$

10. $\sqrt{\frac{x+a}{2x+5}}+|\sin x-2|$, $\frac{9}{2}x+\frac{3}{10}x^2+e^{ctg\,2x}$, $\frac{1}{2}x^{6,3}$

11. $\frac{2x+a}{\sqrt{x}-3}-\log_3|x-4|$, $\frac{1}{2}x+\frac{3}{10}x^2-e^{-tgx+2}$, $\frac{3}{4}x^{3,1}$

12. $\frac{\sqrt{\ln x+2}}{4x-a}+\sin|x+4|$, $\frac{3}{4}x^2-\frac{4}{5}x^3+e^{ctg(x+2)}$, $\frac{3}{5}x^{-0,2}$

13. $\sqrt{\frac{x+1}{2-ax}}-\log_5|x-1|$, $\frac{1}{4}x^2+\frac{7}{5}x+4e^{tg(x-6)}$, $\frac{3}{5}x^{3,4}$

14. $\frac{5x-a}{1-\sqrt{x+1}}-ctg|x-5|$, $\frac{3}{4}x+\frac{6}{5}x^3-7e^{x-8}$, $\frac{1}{4}x^{4,2}$

15. $\frac{2x-a}{3-\sqrt{x-1}}-3\cos|x|$, $\frac{3}{2}x^3-\frac{3}{10}x+7e^{-2tgx}$, $\frac{4}{5}x^{6,8}$

16. $\sqrt{\frac{x+2}{2-ax}}+4|\log_2 x-1|$, $\frac{5}{4}x^2+\frac{3}{10}x-2e^{3tgx+5}$, $\frac{2}{5}x^{2,6}$

17. $\frac{\sqrt{\ln(x+1)}}{5x-a}+\sin|x+5|$, $\frac{7}{2}x^2-\frac{2}{5}x^3+2e^{tgx}$, $\frac{3}{8}x^{-3,8}$

18. $\sqrt{\frac{4x-a}{x-2}}+\log|x-2|$, $\frac{5}{2}x+\frac{7}{5}x^2-e^{4tgx}$, $\frac{3}{4}x^{1,9}$

19. $\sqrt{\frac{2x+a}{4x-3}}+|\cos x|$, $\frac{3}{4}x^3+\frac{9}{5}x+4e^{tg(x-6)}$, $\frac{3}{5}x^{3,3}$

20. $\frac{\sqrt{x+2}}{a-3x}+tg|\cos x+3|$, $\frac{1}{10}x+\frac{5}{4}x^2-e^{2+x}$, $\frac{3}{8}x^{-5,4}$

21. $\frac{x-3a}{2+\sqrt{x-3}}-tg|x-2|$, $\frac{7}{2}x^3-\frac{3}{10}x+e^{ctgx}$, $\frac{2}{5}x^{3,8}$

22. $\frac{\sqrt{x+2}}{2-ax}+|\cos x+5|$, $\frac{3}{4}x^3+\frac{7}{10}x^2+7e^{2-tgx}$, $\frac{7}{8}x^{-0,26}$

23. $\sqrt{\frac{x-a}{2x+1}}+|\sin x-3|$, $\frac{1}{4}x+\frac{11}{5}x^2+e^{tgx-5}$, $\frac{1}{5}x^{3,9}$

24. $\sqrt{\frac{5x+a}{x-3}}+\sin|x-5|$, $\frac{3}{8}x^3-\frac{7}{5}x-6e^{ctgx-1}$, $\frac{2}{5}x^{4,9}$

25. $\frac{x-3}{1-\sqrt{x+a}}-5\sin|x|$, $\frac{1}{4}x-\frac{1}{2}x^3-e^{tgx+7}$, $\frac{5}{8}x^{6,2}$

26. $\sqrt{\frac{x-3}{3x+a}}-5\sin|x|$, $\frac{1}{2}x^2+\frac{8}{5}x-5e^{ctg(x+1)}$, $\frac{1}{4}x^{4,6}$

27. $\sqrt{\frac{x-1}{7x+a}}+|x-tgx|$, $\frac{1}{2}x+\frac{7}{5}x^3+2e^{-x+5}$, $\frac{1}{2}x^{-0,4}$

28. $\frac{\sqrt{3x+5}}{x-a}+\cos|x+5|$, $\frac{1}{10}x^2+\frac{7}{4}x+7e^{ctg(x-3)}$, $\frac{3}{4}x^{-3,11}$

29. $\sqrt{\frac{x+a}{5x-3}}+|\sin x|$, $\frac{3}{2}x^3-\frac{7}{10}x+e^{ctgx}$, $\frac{5}{8}x^{4,8}$

30. $\frac{\sqrt{\ln x+5}}{3x+a}+\log_3|x+5|$, $\frac{1}{8}x^2-\frac{8}{10}x^3+e^{tg3x}$, $\frac{1}{8}x^{-4,4}$

Довідкові матеріали

Структура програми

<table>
<tr><td colspan="3">Директиви препроцесорові</td><td>#include <stdio.h>
#define _USE_MATH_DEFINES
#include <math.h>
#define N 255</td></tr>
<tr><td colspan="3">Прототипи (заголовки) функцій користувача</td><td>int fm(int a, int b);
double fa(double x);</td></tr>
<tr><td colspan="3">Заголовок головної функції (програми)</td><td>int main()</td></tr>
<tr><td colspan="4">{</td></tr>
<tr><td rowspan="2">Тіло функції</td><td>декларативна частина</td><td>визначення констант

опис змінних</td><td>const double A=2.128e-2;
const int B=286;
long int l; unsigned char uc;
int i,j,k; int a=25, h=6;
char str[10];
static int b[2][3]={{1,2,3},{4,5,6}};</td></tr>
<tr><td>виконувана частина</td><td>оператори програми, кожен з яких закінчується ;</td><td>оператор присвоювання a=a+sin(x)-1;
оператор прийняття рішення (умовний) if-else
оператор варіанту switch-case-default
оператор циклу for, while, do-while
оператор функції printf("x=%f\n",x);
scanf("%f",&x);
оператори переходу return, goto, continue, break
блок операторів {…} ({} — операторні дужки)
порожній оператор ; {}</td></tr>
<tr><td colspan="4">}</td></tr>
<tr><td colspan="3">Описи функцій користувача</td><td>int fm(int a,int b) {if (a>b) return a; return b;}
double fa(double x); {if (x<0) x=-x; return x;}</td></tr>
</table>

Ключові (зарезервовані) слова мови С (стандарт С89, або ANSI С),
які не можна використовувати як ідентифікатори*

auto — автоматичний (тип пам'яті); **break** — завершити; **case** — варіант; **char** — символьна величина; **const** — константа **continue** — закінчити поточну ітерацію циклу; **default** — за замовчуванням (дія при відсутності варіанту вибору); **do** — виконати; **double** — дійсна величина подвійної точності; **else** — інакше; **enum** — перераховуваний тип; **extern** — зовнішній (тип пам'яті);	**float** — дійсна величина з плаваючою точкою; **for** — для; **goto** — перейти; **if** — якщо; **int** — цілочисельна величина; **long** — цілочисельна величина подвійної точності; **register** — регістровий (тип пам'яті); **return** — повернення; **short** — цілочисельна величина скороченої довжини; **signed** — ціле число зі знаком (старший розряд вважається знаком); **sizeof** — визначення розміру операнда в байтах;	**static** — статичний (тип пам'яті); **struct** — величина структурного типу (структура); **switch** — перемикач; **typedef** — визначення типу (визначає скорочене ім'я для позначення типу); **union** — об'єднання; **unsigned** — величина беззнакового типу (старший розряд не вважається знаком); **void** — величина, значення якої відсутнє; **volatile** — об'єкт, значення якого може змінюватися без явних вказівок програміста; **while** — доки.

У стандарті С99 додано ще 5 ключових слів: **inline**, **restrict**, **_Bool**, **_Complex**, **_Imaginary**. Ці слова, крім слова **inline** (вказівка компілятору оптимізувати виклик функції), в середовищі VS 2010 не є ключовими.

У стандарті С11 додано ще 7 ключових слів: **_Alignas, _Alignof, _Atomic, _Generic, _Noreturn, _Static_assert, _Thread_local**.

*Але, якщо підімкнуто певну бібліотеку й ім'я змінної збігається з іменем бібліотечної функції, то не можна буде скористатися відповідною функцією підімкнутої бібліотеки.

Типи даних, визначені стандартом С

Тип	Пам'ять (байтів) у VS 2010	Діапазон значень у VS 2010
char / signed char	1	–128 … 127 (-2^7 … 2^7-1)
unsigned char	1	0 … 255 (2^8-1)
short int / signed short int	2	–32768 … 32767 (-2^{15} … $2^{15}-1$)
unsigned short int	2	0 … 65535 ($2^{16}-1$)
int / signed int / long int / signed long int	4	–2147483648 … 2147483647 (-2^{31} … $2^{31}-1$)
unsigned int / unsigned long int	4	0 ... 4294967295 ($2^{32}-1$)
long long int	8	–9223372036854775808 … 9223372036854775807 (-2^{63} … $2^{63}-1$); *стандарт С99*
unsigned long long int	8	0 ... 18446744073709551615 (0 … $2^{64}-1$); *стандарт С99*
float	4	9.9e–45 … 3.4e+38; точність — 6-7 десяткових цифр; мантиса — 24 біти
double / long double	8	4.94e–324 ... 1.7e+308; точність — 15-16 десяткових цифр; мантиса — 53 біти

Перетворення типів операндів у виразах
(перетворення виконуються відповідно до таблиці при перегляді її згори вниз)

№	Початкові типи операндів	Типи операндів після перетворення	Пояснення
1.	long double α ∀ β	long double α **long double β**	якщо один операнд має тип long double, то і другий операнд перетвориться на long double
2.	double α ∀ β	double α **double β**	якщо один операнд має тип double, то і другий операнд перетвориться на double
3.	float α ∀ β	float α **float α**	якщо один операнд має тип float, то і другий операнд перетвориться на float
4.	long long int α ∀ ціле зі знаком β	long long int α **long long int β**	якщо один операнд має тип long long int і другий операнд є беззнаковим цілим, то і другий перетвориться на long long int
5.	unsigned long int α ∀ β	unsigned long int α **unsigned long int β**	якщо один операнд має тип unsigned long int, то і другий перетвориться на unsigned long int
6.	long int α unsigned int β	long int α **long int β**	якщо long int покриває всі значення unsigned int, то другий операнд перетвориться на long int
		unsigned long int α **unsigned long int β**	якщо long int не покриває всі значення unsigned int, то обидва операнди перетворяться на unsigned long int
7.	long int α ∀ β	long int α **long int β**	якщо один операнд має тип long int, то і другий перетвориться на long int
8.	unsigned int α ∀ β	unsigned int α **unsigned int β**	якщо один операнд має тип unsigned int, то і другий перетвориться на unsigned int
9.	char α	**int α**	усі операнди типу char перетворюються на int
10.	unsigned char α		
11.	short int α	**int α**	усі операнди типу short int перетворюються на int
12.	unsigned short int α	**unsigned int α**	усі операнди типу unsigned short int перетворюються на unsigned int

Із наведених у таблиці правил перетворення видно, що якщо операнди мають різні типи, то відбувається приведення операнда молодшого типу до старшого типу; при цьому для даних цілих типів старшим вважається тип unsigned порівняно з типом signed. Проте, хоч стандартом мови С визначається перетворення операнда до типу unsigned, реалізація може залежати від компілятора. Поєднання у виразах цілих чисел типів signed і unsigned може призвести до непередбачуваних результатів. Тому у виразах краще використовувати явне перетворення типів.

Операції

(операції подано за спаданням пріоритетів;
у кожній групі, відокремленій суцільною лінією, пріоритети однакові):

Група операцій	Знак операції	Опис
первинні	( )	підвищення пріоритету (дужки)
	[ ]	виділення елемента масиву
	.	виділення елемента запису
	->	виділення елемента запису
унарні	!	логічне заперечення
	~	побітове заперечення
	-	зміна знака
	++	збільшення на одиницю
	--	зменшення на одиницю
	&	взяття адреси
	*	звернення за адресою
	(тип)	перетворення типу
	sizeof( )	визначення розміру в байтах
мультиплікативні	*	множення
	/	ділення
	%	визначення остачі від ділення
адитивні	+	додавання
	-	віднімання
зрушення	<<	зсув вліво
	>>	зсув вправо
відношень (порівняння)	<	менше ніж
	<=	менше або дорівнює
	>	більше ніж
	>=	більше або дорівнює
	==	дорівнює
	!=	не дорівнює
побітові (порозрядні)	&	побітове логічне «і» (логічне множення)
	^	побітове виключне «або»
	|	побітове логічне «або»
логічні	&&	логічне «і»
	||	логічне «або»
умови	?:	умовна (тернарна) операція
присвоювання	=	присвоювання
	+=, - =, *=, /=, %=, <<=, >>=, &=, |=, ^=	складені операції присвоювання (a*=b означає a=a*b)
кома	,	операція кома

Основні стандартні математичні функції і константи мови С

Аргументи функцій записують в круглих дужках. Наприклад: **`sin(x)`**, **`abs(x-1)`**, **`fabs(sin(a))`** тощо.

При обчисленні значення функції x^y (реалізує функція pow(x,y)), якщо $x > 0$, можна використовувати рівність $x^y = e^{y \ln x}$ (exp(y*log(x)).

Значення $\log_a b$ обчислюється за формулою $\log_a b = \dfrac{\ln b}{\ln a}$.

Значення $\operatorname{ctg}\alpha$ обчислюється за формулою $\operatorname{ctg}\alpha = \dfrac{\cos\alpha}{\sin\alpha}$ або $\operatorname{ctg}\alpha = \dfrac{1}{\operatorname{tg}\alpha}$.

Значення числа $e = 2{,}718281828459045$ можна одержати, скориставшись функцією e^x: $e = \exp(1)$.

Якщо в тексті програми курсор навести на ім'я функції, то відобразяться всі можливі її прототипи.

Якщо в тексті програми в директиві #include <xxxxx.h> клацнути правою клавішею мишки по імені бібліотеки і в контекстному меню вибрати команду Open Document <xxxxx.h>, то відкриється заголовочний файл бібліотеки, в якому за допомогою пошуку (Ctrl+F) можна знайти потрібну інформацію.

Бібліотеки мови С, їхні функції й константи описано на сайті www.cplusplus.com/reference/

а) бібліотека math.h* (www.cplusplus.com/reference/cmath/ldexp/) — **функції і константи**

Ім'я функції**	**Прототип**	**Опис**
sin	double sin(double x);	синус $\sin x$
cos	double cos(double x);	косинус $\cos x$
tan	double tan(double x);	тангенс $\operatorname{tg} x$
asin	double asin(double x);	арксинус $\arcsin x$
acos	double acos(double x);	арккосинус $\arccos x$
atan	double atan(double x);	арктангенс $\operatorname{arctg} x$
sinh	double sinh(double x);	гіперболічний синус $\operatorname{sh} x = (e^x - e^{-x})/2$
cosh	double cosh(double x);	гіперболічний косинус $\operatorname{ch} x = (e^x + e^{-x})/2$
tanh	double tanh(double x);	гіперболічний тангенс $\operatorname{th} x = \operatorname{sh} x / \operatorname{ch} x$
sqrt	double sqrt(double x);	квадратний корінь $\sqrt{x}$
cbrt	double cbrt(double x);	кубічний корінь $\sqrt[3]{x}$; *стандарт С99*
pow(x,y)	double pow(double x, double y); або double pow(double x, int y);	значення x^y
exp	double exp(double x);	показникова функція e^x
log	double log(double x);	натуральний логарифм $\ln x$
log10	double log10(double x);	десятковий логарифм $\log_{10} x$
abs	int abs(double x); або int abs(int x);	ціле абсолютне значення $\lvert x \rvert$
fabs	double fabs(double x);	абсолютне значення $\lvert x \rvert$
round	double round(double x);	заокруглює до **найближчого цілого** числа (**`round(3.1)=3; round(3.5)=4; round(3.8)=4;round(-3.1)= -3; round(-3.5)= -4; round(-3.8)= -4`**); *стандарт С99*
trunc	double trunc(double x);	відкидає дробову частину числа (**`trunc(3.1)=3; trunc(3.5)=3; trunc(3.8)=3; trunc(-3.1)= -3; trunc(-3.5)= -3; trunc(-3.8)= -3`**); *стандарт С99*
ceil	double ceil(double x);	заокруглює до **найближчого більшого цілого** числа (**`ceil(3.1)=4; ceil(3.5)=4; ceil(3.8)=4; ceil(-3.1)= -3;`**

		ceil(–3.5)= –3; ceil(–3.8)= –3)
floor	double floor(double x);	заокруглює до **найближчого меншого цілого** числа (**floor(3.1)=3; floor(3.5)=3; floor(3.8)=3;floor(–3.1)= –4; floor(–3.5)= –4; floor(–3.5)= –4**)
fmod	double fmod(double x, double y);	залишок від ділення двох чисел x/y (**fmod(3.1, 3.8)=3.1; fmod(3.1, 1.4)=0.3; fmod(–3.1, 1.4)= –0.3**)
fmax	double fmax(double x, double y);	максимальне значення $\max(x, y)$; *стандарт С99*
fmin	double fmin(double x, double y);	мінімальне значення $\min(x, y)$; *стандарт С99*

*Бібліотека <tgmath.h> (стандарт С99) дає можливість виконувати всі функції бібліотеки <math.h>; при цьому аргументи функцій можуть бути як дійсними, так і цілими. Проте в VS 2010 цієї бібліотеки нема.

**Згідно зі стандартом С99 для роботи з типами float і long double існують функції-відповідники з постфіксами f і l до всіх вказаних у таблиці функцій (наприклад, поданій у таблиці функції sin — double sin(double x) відповідають функції sinf — float sinf(float x) і sinl — long double sinl(long double x)).

Усі функції, які одержують або повертають значення кута, працюють з радіанами (наприклад, 180° відповідає число $\pi \approx 3{,}14159265$; 90° — $\pi/2 \approx 1{,}57079632$; 45° — $\pi/4 \approx 0{,}78539816$; 30° — $\pi/6 \approx 0{,}52359878$). Для переходу від градусної міри до радіанної треба виконати перетворення $\varphi_r = \varphi_\circ \cdot \pi/180$; навпаки — $\varphi_\circ = \varphi_r \cdot 180/\pi$.

Ім'я константи*	Значення в середовищі VS 2010	Опис
M_E	2,71828182845904523536	число e
M_LOG2E	1,44269504088896340736	$\log_2 e$
M_LOG10E	0,434294481903251827651	$\log_{10} e$
M_LN2	0,693147180559945309417	$\ln 2$
M_LN10	2,30258509299404568402	$\ln 10$
M_PI	3,14159265358979323846	число π
M_PI_2	1,57079632679489661923	$\pi/2$
M_PI_4	0,78539816339744830962	$\pi/4$
M_1_PI	0,318309886183790671538	$1/\pi$
M_2_PI	0,636619772367581343076	$2/\pi$
M_2_SQRTPI	1,12837916709551257390	$2/\sqrt{\pi}$
M_SQRT2	1,41421356237309504880	$\sqrt{2}$
M_SQRT1_2	0,707106781186547524401	$1/\sqrt{2}$

*Щоб скористатися цими константами, треба препроцесорові вказати директиву `#define _USE_MATH_DEFINES` ще до директиви підмикання бібліотеки `#include <math.h>`

б) бібліотека stdlib.h (http://www.cplusplus.com/reference/cstdlib/) — **функції**

Ім'я функції	Прототип	Опис
abs labs llabs	int abs (int n); long int labs (long int n); long long int llabs (long long int n);	абсолютне значення* $\lvert x \rvert$
srand	void srand (unsigned int seed);	ініціалізація генерації випадкових чисел (**#include <time.h> /*…*/ srand(time(0));**)
rand	int rand (void);	генерація випадкових чисел (у VS 2010 максимальне згенероване число визначається константою RAND_MAX = 32767, мінімальне — 0; **rand()%11** — від 0 до 10; **rand()%3–1** — –1, 0 чи 1)

*У середовищі VS 2010 функції abs, labs і llabs розміщуються також і в бібліотеці math.h. Функція abs з бібліотеки math.h дає можливість працювати з даними типу int, long int і long long int. Також функція abs з бібліотеки math.h може працювати з дійсними аргументами, але результатом при цьому є ціле значення (дробова частина відкидається). Функція abs з бібліотеки stdlib.h аргументом може мати тільки ціле значення.

в) бібліотека limits.h (www.cplusplus.com/reference/climits/) — **мінімальні й максимальні цілі значення**

Ім'я константи	Значення в середовищі VS* і формат виведення функцією printf	Опис
SHRT_MIN	–32768 (-2^{15}) %d або %hd	мінімальне значення об'єкта типу short int
SHRT_MAX	32767 ($2^{15}-1$) %d або %hd	максимальне значення об'єкта типу short int
USHRT_MAX	65535 ($2^{16}-1$) %d, або %hu, або %u	максимальне значення об'єкта типу unsigned short int
INT_MIN	–2147483648 (-2^{31}) %d	мінімальне значення об'єкта типу int
INT_MAX	2147483647 ($2^{31}-1$) %d або %u	максимальне значення об'єкта типу int
UINT_MAX	4294967295 ($2^{32}-1$) %u	максимальне значення об'єкта типу unsigned int
LONG_MIN	–2147483648 (-2^{31}) %d або %ld	мінімальне значення об'єкта типу long int
LONG_MAX	2147483647 ($2^{31}-1$) %d, або %ld, або %u	максимальне значення об'єкта типу long int
ULONG_MAX	4294967295 ($2^{32}-1$) %u або %lu	максимальне значення об'єкта типу unsigned long int
LLONG_MIN	–9223372036854775808 (-2^{63}) %lld	мінімальне значення об'єкта типу long long int
LLONG_MAX	9223372036854775807 ($2^{63}-1$) %lld	максимальне значення об'єкта типу long long int
ULLONG_MAX	18446744073709551615 ($2^{64}-1$) %llu	максимальне значення об'єкта типу unsigned long long int

*Значення залежить від конкретної системи і реалізації бібліотеки

г) бібліотека float.h (www.cplusplus.com/reference/cfloat/) — **мінімальні й максимальні дійсні значення**

Ім'я константи	Значення в середовищі VS 2010* і формат виведення функцією printf	Опис
DBL_EPSILON LDBL_EPSILON	2,2204460492503131e–016 %e або %g	найменше додатне значення типу double чи long double таке, що 1,0+DBL_EPSILON ≠1,0
DBL_MAX LDBL_MAX	1,7976931348623158e+308 %f, або %e, або %g	максимальне значення типу double чи long double
DBL_MIN LDBL_MIN	2,2250738585072014e–308 %e або %g	мінімальне додатне значення типу double чи long double
FLT_EPSILON	1,192092896e–07F %e або %g	найменше додатне значення типу float таке, що 1,0+ FLT_EPSILON ≠1,0
FLT_MAX	3,402823466e+38F %f, або %e, або %g	максимальне значення типу float
FLT_MIN	1,175494351e–38F %e або %g	мінімальне додатне значення типу float

Основні стандартні функції мови С для введення й виведення інформації

а) бібліотека stdio.h (www.cplusplus.com/reference/cstdio)**, яка реалізує основні можливості введення й виведення інформації**

Ім'я функції	Прототип	Опис
printf	int printf(const char * format, ...);	форматований вивід даних у стандартний потік виведення (stdout)
scanf	int scanf(const char * format, ...);	форматований ввід даних із стандартного потоку введення (stdin)
getchar	int getchar(void);	ввід наступного символу із стандартного потоку введення; чекає натискання клавіші Entet
gets	char * gets(char * str);	ввід рядка символів (можуть бути пробіли) із стандартного потоку введення; чекає натискання клавіші Entet
fflush	int fflush(FILE * stream);	якщо потік відкрито для виводу, то із буфера дані записуються в зв'язаний з потоком файл; якщо потік відкрито для вводу, то буфер очищається
flushall	int flushall(void)	очистка всіх вхідних буферів і запис фізичного вмісту всіх вихідних буферів, зв'язаних з потоками файлів, у відповідні їм файли; усі потоки залишаються відкритими; *не визначена стандартом С*

Функція printf

Функція printf повертає кількість реально виведених символів (у випадку помилки — від'ємне значення). Звернення до функції:

printf ("рядок формату", список аргументів);

Аргументи у списку відокремлюються комами; аргументами можуть бути вирази у широкому розумінні (вирази, звернення до функцій, змінні, константи). Рядок формату містить специфікатори форматів і символи для виведення.

Специфікатор формату функції printf має вигляд (квадратні дужки не друкуються — вони лише вказують, що параметр не обов'язковий):

% [модифікатори] код_формату

Специфікації формату зіставляються з аргументами відповідно до порядку в списку аргументів (наприклад, `printf("i=%d - %c",325,'?');` дає `i=325 - ?`; `printf("%-+7.5d=i",325);` дає `+00325 =i`).

Коди форматів, які означають, що значенням аргумента є:

d або **i** — десяткове ціле число зі знаком (ці коди еквівалентні);
u — беззнакове десяткове ціле число;
o — вісьміркове ціле число без знака;
x або **X** — шістнадцяткове ціле число без знака з цифрами 0,..., 9, a, b, c, d, e, f або 0,..., 9, A, B, C, D, E, F відповідно;
f — дійсне десяткове число з фіксованою точкою;
e або **E** — дійсне десяткове число в експоненційній формі виду 1.23457e+002 або 1.23457E+002 відповідно;
g або **G** — використовується, як і код f чи e; незначущі нулі не виводяться; залежно від того, яке подання числа буде коротшим, використовується код f чи e (якщо менше чотирьох значущих нулів, то f);
c — символ;
s — рядок символів (символи рядка виводяться до символа кінця рядка або доки не буде виведено кількість символів, задану точністю);
p — вказівник (виводиться машинна адреса, формат якої сумісний з типом адресації комп'ютера);
n — ніяка інформація на екран не виводиться; аргумент, який відповідає цьому коду, має бути вказівником на цілочисельну змінну (у цю змінну записується кількість виведених символів).

Модифікатори (не обов'язкові; подано в порядку їхнього використання):

мінус (**-**) — вирівнювання значеня по лівому краю у своєму полі (ставиться зразу після %; якщо мінуса нема — по правому);
плюс (+) — число виводиться зі знаком;
ціле число — специфікація мінімальної ширини поля (зайві позиції заповнюються пробілами; якщо довжина числа чи рядка символів більша, то значення виводиться повністю);
ціле число з нулем попереду — специфікація мінімальної ширини поля (зайві позиції заповнюються нулями; якщо довжина числа чи рядка символів більша, то значення виводиться повністю);

крапка і ціле число — модифікатор точності вказується після специфікації мінімальної ширини. Для форматів e, E, f — кількість цифр після коми; g чи G — максимальна кількість значущих цифр; d чи i — мінімальна кількість цифр і в разі необхідності перед числом буде додано нулі; s — максимальна довжина поля (якщо рядок буде довшим, то кінцеві символи не виведуться);

зірочка (*) — значення специфікації мінімальної ширини поля і модифікатора точності можна задати не константами, а аргументами функції, вказавши зірочку, яка зіставляється з відповідним аргументом у списку аргументів;

h — визначає аргумент типу short int і записується перед кодами форматів d, i, o, u, x, X; перед кодом n — вказує, що відповідний аргумент має тип short int;

l — визначає аргумен типу long int (записується перед кодами форматів d, i, u, o, x, X) чи long double (стандарт C99; записується перед кодами форматів f, e, E, g, G, але **не дає ніякого ефекту**); перед кодом n — вказує, що відповідний аргумент має тип long int; перед кодом c — вказує, що відповідний аргумент є символом в розширеному 16-бітовому алфавіті;

L — визначає аргумен типу long double (записується перед кодами форматів f, e, E, g, G);

hh (стандарт C99) — визначає аргумен типу signed чи unsigned char і записується перед кодами форматів d, i, u, o, x, X або вказівник на змінну типу signed char для коду n.

ll (стандарт C99) — визначає аргумен типу signed чи unsigned long long int і записується перед кодами форматів d, i, u, o, x, X або вказівник на змінну типу long long int для коду n;

F і **N** —визначають відповідно вказівники типу far і near;

— записується перед кодами форматів f, e, E, g, G, щоб завжди виводити десяткову точку; перед x чи X —виводити префікс 0x чи 0X перед шістнадцятковим поданням числа; перед o — виводити префікс 0 перед вісьмірковим поданням числа;

якщо після % записано не символ перетворення, то він виводиться на екран (якщо задати %%, то на екран буде виведено символ %).

Найчастіше використовувані *керуючі символьні константи*, які мають зарезервовані позначення:

\n — перехід на новий рядок;
\t — горизонтальна табуляція (8 позицій);
\r — повернення курсора на початок поточного рядка;
\b — повернення курсора на на одну позицію вліво (назад) в поточному рядку;
\a —короткий звуковий сигнал;
\\ — виведення символа \ ;
\' — виведення символа ' ;
\" — виведення символа " ;
\? — виведення символа ? (символ “?” потрапив до зарезервованих символів у зв’язку з тим, що в C є операція, яка позначається знаком “?”; у деяких ситуаціях це могло б призвести до неоднозначного тлумачення конструкцій мови. Символьну константу “знак питання” можна записати будь-яким способом: '?', '\77', '\x3F' чи '\?').

Для переносу тексту на наступний рядок використовується символ \:

```
printf ("AAAA\   /*початок тексту*/
BBB\n "); /*закінчення тексту*/
printf("AAAABBB\n"); /*еквівалентний оператор*/
```

Функція scanf

Функція scanf повертає кількість реально введених і присвоєних змінним елементів даних (полів). При виявленні помилки до присвоєння значення першого поля функція повертає значення EOF. Звернення до функції:

scanf ("рядок формату", список аргументів);

Усі аргументи повинні бути вказівниками на змінні — адресами змінних (наприклад, **`&aaa, &I, str`**, де **`str`** — рядок символів). Рядок формату містить символи трьох категорій: специфікатори форматів, пробільні символи і символи, відмінні від пробільних.

Специфікатор формату функції scanf має вигляд (модифікатори не обов’язкові):

% [модифікатори] код_формату

Специфікації формату зіставляються з аргументами відповідно до їхнього порядку в списку аргументів (наприклад: **`scanf("%d%f%c%s",&a,&b,&ch,str);`** — введення цілого і дійсного чисел, символа і рядка; або **`scanf("%d %f  %c  %10s",&a,&b,&ch,str);`** — пробіли між форматами ролі не грають).

Коди форматів, які означають, що зчитуваним значенням аргумента є:

d — десяткове ціле число зі знаком;
i —ціле число зі знаком у будь-якому форматі (десяткове, вісьміркове, шістнадцяткове);
u — беззнакове десяткове ціле число;
f або **F** (стандарт C99)— дійсне десяткове число типу float з фіксованою точкою;

e або **E** — дійсне десяткове число типу float в експоненційній формі (при введенні символ E має бути на тому ж регістрі, що й код);
g або **G** — дійсне десяткове число типу float;
o — вісьміркове ціле число без знака;
x або **X** — шістнадцяткове ціле число без знака (при введенні цифри від A до F мають бути на тому ж регістрі, що й код);
c — один символ;
s — рядок символів (до першого пробільного символа; інакше — застосовувати функцію gets);
p — вказівник;
n — набуває цілого значення, рівного кількості прочитаних досі символів;
%% — читає знак процента.

Модифікатори (не обов'язкові)

l — перед кодами f, e, E, g, G для зчитування даних у змінні типу double; перед c, s — для зчитування у двобайтові символи чи рядки двобайтових символів (для зчитування даних типу double %f не підходить — підходить %lf);
L — перед кодами f, e, E, g, G для зчитування даних у змінні типу long double;
hh (стандарт C99) — визначає аргумен типу signed чи unsigned char і записується перед кодами форматів d, i, u, o, x, X або вказівник на змінну типу signed char для коду n.
ll (стандарт C99) — визначає аргумен типу signed чи unsigned long long int і записується перед кодами форматів d, i, u, o, x, X або вказівник на змінну типу long long int для коду n;
ціле число — модифікатор максимальної довжини поля (ставиться між знаком % і кодом формату) обмежує кількість символів, які зчитуються; якщо роздільник буде зчитано раніше, ніж досягнуто вказаної максимальної довжини поля, то введення даних припиниться;
зірочка (*) —читає, але не присвоює дані заданого типу (ставиться між знаком % і кодом формату);

Пробільні символи — роздільники (якщо в рядку форматування поточним символом є роздільник, то у вхідному потоці пропускається один чи кілька роздільників до першого символа, відмінного від роздільника):

пробіли;
символи табуляції;
роздільники рядків (Enter);
символи кома, крапка з комою тощо не є роздільниками.

Не зважаючи на те, що пробіли, символи табуляції і роздільники рядків використовуються як роздільники полів зчитування, при зчитуванні окремого символа (формат %c) вони читаються, як і будь-який інший символ.

Символи, *відмінні від пробільних* — якщо в рядку форматування поточним символом є символ, відмінний від роздільника, то функція прочитає і відкине аналогічний символ у вхідному потоці; якщо вказаного символа у вхідному потоці нема, то функція закінчує роботу.

б) бібліотека stdlib.h (www.cplusplus.com/reference/stdlib)

Ім'я функції	Прототип	Опис
system	int system(const char* command);	виконання системної команди (**`system("pause")`** — чекає натискання будь-якої клавіші на клавіатурі; **`system("cls");`** — очищає екран; **`system("chcp 1251")`** — установка кодової сторінки win-cp 1251 (кодування ANSI) в потік вводу і виводу (change codepage); **`system("chcp 1251 & cls"); system(NULL); system("dir"); system("color 0B");`**)