

Лабораторна робота 1

ДОСЛІДЖЕННЯ ФУНКЦІЙ ТА ПРОЦЕДУР (МЕТОДІВ) У МОВАХ C++ ТА C#

Мета лабораторної роботи - вивчити устрій і правила написання функцій та процедур (методів) в мовах C++ та C#, способи передачі параметрів у функцію та процедуру і повертання результату.

Теоретичні відомості

...

Завдання

Написати функції на мовах C++ та C#, які моделюють на бітовому рівні операції згідно з варіантом. Для перевірки вірності роботи програм використати значення з варіанта. Перша функція завдання повертає результат через параметр, друга - оператором return.

Таблиця варіантів

№ варіанта	Операція	Операнд	Значення для перевірки		
1	Лічильник збільшення на 1	1	31	56	-25
	Операція відношення >	1	44	46	15
		2	44	-7	30
2	Лічильник зменшення на 1	1	16	61	-37
	Операція відношення ==	1	100	-8	132
		2	100	125	131
3	Операція відношення <	1	84	-77	63
		2	63	12	-34
	Операція додавання +	1	-10	15	31
		2	12	41	32
4	Лічильник збільшення на 1	1	63	-18	92
	Операція відношення >	1	88	34	23
		2	88	2	57
5	Лічильник збільшення на 1	1	63	-18	92
	Операція відношення ==	1	11	84	61
		2	11	-2	107
6	Лічильник зменшення на 1	1	-255	94	46
	Операція додавання +	1	-52	31	111
		2	18	32	49
7	Лічильник збільшення на 1	1	-87	511	183
	Операція відношення <	1	100	-8	132
		2	100	125	131
8	Лічильник зменшення на 1	1	-24	128	13
	Операція відношення <	1	0	36	16

№ варіанта	Операція	Операнд	Значення для перевірки		
		2	-7	136	16
9	Лічильник збільшення на 1	1	-128	45	63
	Операція відношення <	1	33	-6	26
		2	117	-10	26
10	Лічильник зменшення на 1	1	0	-15	138
	Операція відношення >	1	-1	94	537
		2	-12	305	537
11	Операція відношення =	1	-57	124	1
		2	-57	-10	2
	Операція додавання +	1	82	229	-3
		2	46	110	151
12	Операція додавання +	1	141	-12	173
		2	115	205	347
	Операція відношення =	1	-21	79	115
		2	-21	80	41
13	Лічильник збільшення на 1	1	-43	75	63
	Операція відношення =	1	48	69	24
		2	48	78	25
14	Лічильник зменшення на 1	1	-92	16	62
	Операція відношення <	1	54	38	143
		2	54	40	129
15	Операція відношення >	1	211	183	86
		2	211	182	134
	Операція додавання +	1	69	48	53
		2	-73	24	19

Методичні вказівки

Для моделювання на бітовому рівні арифметичних операцій та операцій відношення необхідно вміти вирішувати наступні задачі:

- визначати значення окремих бітів операнду(*i*в);
- формувати результат шляхом встановлення в 0 чи 1 певних бітів результуючого значення.

При цьому, реалізація операцій відношення потребує вирішення лише першої задачі, тоді як реалізація арифметичних операцій вимагає виконання обох задач.

Для визначення значення окремого *n*-го біту (`sizeof(тип значення) >= n >= 0`) операнду необхідно використовувати маску, що дорівнює 1, та бітові операції «І» та «зсув ліворуч». Наприклад, щоб дізнатись значення *p*'ятого біту вмісту цілої змінної *iA* необхідно виконати наступні перетворення:

$$iBitA = iA \text{ I } (1 \text{ зсув ліворуч } 5).$$

Якщо значення $iBitA$ дорівнюється 0, п'ятий біт числа iA також є рівним 0, якщо будь-якому додатному значенню, біт дорівнюється 1.

Порівнюючи значення відповідних бітів заданих операндів, можна визначити за їхнім бітовим представленням, який з операндів є більшим, меншим, тощо.

Наприклад:

Операція відношення $=$

$$\begin{aligned} iA &= 25_{10} = \underline{1}1001_2 & iBitA &= iA \text{ та } (1 \text{ зсув ліворуч } 4) = 16 \Rightarrow \underline{1} \\ iB &= 9_{10} = \underline{0}1001_2 & iBitB &= iB \text{ та } (1 \text{ зсув ліворуч } 4) = 0 \Rightarrow \underline{0} \end{aligned} \quad \Rightarrow 1 < 0 \Rightarrow iA < iB$$

Для вирішення завдання моделювання операцій додавання та лічильника недостатньо вміти визначати значення окремих бітів операндів, але потрібно формувати результуюче значення, використовуючи його бітове представлення. Для цього використовуються маска та бітові операції «зсув ліворуч», «АБО», «АБО, що виключає»:

Наприклад:

Операція додавання $+$

$$\begin{aligned} iA &= 25_{10} = 11001_2 \\ iB &= 9_{10} = 01001_2 \\ iSum &= 34_{10} = \underline{1}000\underline{1}0_2 \end{aligned}$$

Результат $iSum$ формується встановленням в 1 першого та п'ятого бітів, тобто:

$$\begin{aligned} 1) iSum &= 0, & iSum &= 0_{10} = 0_2 \\ 2) iSum &= iSum \text{ АБО } (1 \text{ зсув ліворуч } 1), & iSum &= 2_{10} = 000\underline{1}0_2 \\ 3) iSum &= iSum \text{ АБО } (1 \text{ зсув ліворуч } 5), & iSum &= 34_{10} = \underline{1}000\underline{1}0_2 \end{aligned}$$

Лічильник, або операції зменшення чи збільшення на одиницю не зводяться до операції додавання. Для отримання результату лічильника необхідно використовувати певні закономірності перетворення значень та застосовувати їх при побудові алгоритму для вирішення задачі.

Наприклад, при зменшенні числа на одиницю в результуючому значенні всі біти, що дорівнюють 0, починаючи з молодшого, та перший слідуючий за ними біт, що дорівнюється 1, інвертуються, а старші залишаються незмінними. Для інвертування відповідних бітів використовується операція «АБО, що виключає» (далі «АБОВикл»).

Наприклад:

Лічильник зменшення на 1

$$\begin{aligned} iA &= 25_{10} = 11001_2 \\ iA - 1 &= 24_{10} = 1100\underline{0}_2 \end{aligned}$$

$$1) iA = iA \text{ АБОвикл (1 зсув ліворуч 0),} \quad iA = 24_{10} = 1100\textbf{0}_2$$

$$iA = 24_{10} = 11000_2$$

$$iA - 1 = 23_{10} = \textbf{10111}_2$$

$$1) iA = iA \text{ АБОвикл (1 зсув ліворуч 0),} \quad iA = 25_{10} = 1100\textbf{1}_2$$

$$2) iA = iA \text{ АБОвикл (1 зсув ліворуч 1),} \quad iA = 27_{10} = 110\textbf{11}_2$$

$$3) iA = iA \text{ АБОвикл (1 зсув ліворуч 2),} \quad iA = 31_{10} = 11\textbf{111}_2$$

$$4) iA = iA \text{ АБОвикл (1 зсув ліворуч 3),} \quad iA = 23_{10} = \textbf{10111}_2$$

Контрольні запитання

1. Поясніть призначення функції (процедури).
2. Наведіть відмінності процедури та функції.
3. Що таке прототип функції (процедури)?
4. Наведіть способи передачі параметрів у функції, надайте приклади на мовах C++ та C#.
5. Наведіть способи повертання результатів з функції, надайте приклади на мовах C++ та C#.
6. Що таке локальні змінні функції (процедури)?