

**Міністерство освіти і науки України**  
**Національний університет “Львівська політехніка”**



**Лабораторна робота №9**  
**з дисципліни «Програмування**  
**частина 2»**

**Виконав:**  
Студент групи АП-11  
Братейко Вадим

**Прийняв:**  
Чайковський І.Б.

Львів 2024

## «Логічні та побітові операції у мові С»

**Мета роботи:** навчитися використовувати логічні та побітові операції під час програмування на мові С.

Теоретичні відомості В програмуванні треба мати можливість не лише проводити обчислення над числовими даними, тобто робити арифметичні операції, але й обробляти логічні дані. З логічними даними програма має справу, коли перевіряє чи виконується деяка умова

В мові С логічні значення зображуються за допомогою цілих чисел. А саме, число 0 зображує логічну хибу, а будь-яке відмінне від нуля число зображує логічну істину.

В мові С існує три логічні операції:

1. Логічна операція І&&;
2. Логічна операція АБО ||;
3. Логічна операція НЕ ! або логічне заперечення.

Операції	Позначення	Умова	Короткий опис
І	&&	$a==3 \ \&\& \ b>4$	Складена умова істинна, якщо істинні обидві прості умови
АБО		$a==3 \    \ b>4$	Складена умова істинна, якщо істинна, хоча б одна з простих умов
НЕ	!	$! (a==3)$	Умова істинна, якщо а не дорівнює 3

## Операції порівняння

Операція	Значення
<	менше
<=	менше або рівне
==	перевірка на рівність
>=	більше або рівне
>	більше
!=	перевірка на нерівність

### Пріоритет операцій в С

Пріоритет	Операція	Асоціативність	Опис
1	::	зліва направо	унарна операція дозволу області дії
	[]		операція індексування
	()		круглі скобки
	.		звернення до члена структури або класу
	->		звернення до члена структури або класу через покажчик
2	++	зліва направо	постфіксний інкремент
	--		постфіксний декремент
3	++	справа наліво	префіксний інкремент
	--		префіксний декремент
4	*	зліва направо	множення
	/		ділення
	%		залишок від ділення
5	+	зліва направо	додавання
	-		віднімання
6	>>	зліва направо	зсув вправо
	<<		зрушення вліво
7	<	зліва направо	менше
	<=		менше або дорівнює
	>		більше
	>=		більше або дорівнює
8	=	зліва направо	дорівнює
	!=		не дорівнює
9	&&	зліва направо	Логічне І
10		зліва направо	Логічне АБО
11	?:	справа наліво	умовна операція (тернарного операція)
12	=	справа наліво	присвоювання
	*=		множення з привласненням
	/=		поділ з привласненням
	%=		залишок від ділення з привласненням
	+=		додавання з привласненням
	-=		віднімання з привласненням
13	,	зліва направо	кома

#### Приклад 1

```
#include <stdio.h>
```

```
int main() {
```

```
    int a = 017; // 017 відповідає 15 у десятковій системі
```

```
    int b = 036; // 036 відповідає 30 у десятковій системі
```

```
    // Побітове І
```

```
    int bitwise_and = a & b;
```

```
    printf("a & b = %o\n", bitwise_and); // %o для виводу у вісімковій системі
```

```
    // Побітове АБО
```

```
    int bitwise_or = a | b;
```

```
    printf("a | b = %o\n", bitwise_or);
```

```
// Зсув вправо на 2 (тільки для змінної a)
int left_shift_a = a >> 2;
printf("a >> 2 = %o\n", left_shift_a);
```

```
a & b = 16
a | b = 37
a << 2 = 3
}
```

## Приклад 2

1) Переведення значень змінних  $a = 017$ ,  $b = 036$  з вісімкової у двійкову систему числення:

$a = 017$  (вісімкова) = 000 001 111 (двійкова)

$b = 036$  (вісімкова) = 011 110 (двійкова)

2) Виконання необхідних операцій:

\*Побітове І ( $a \& b$ ):

```
000 001 111
& 000 011 110
000 001 110
```

\*Побітове АБО ( $a | b$ ):

```
000 001 111
| 000 011 110
000 011 111
```

\*Зсув вліво на 2 (тільки для  $a$ ) ( $a \ll 2$ ):

$000 001 111 \ll 2 = 000 111 100$

\*Зсув вправо на 2 (тільки для  $a$ ) ( $a \gg 2$ ):

$000 001 111 \gg 2 = 000 000 011$

3) Отже, результати операцій для змінних  $a$  та  $b$ :

$a \& b = 000 001 110$  (вісімкова: 016)

$a | b = 000 011 111$  (вісімкова: 037)

$a \ll 2 = 000 111 100$  (вісімкова: 074)

$a \gg 2 = 000 000 011$  (вісімкова: 003)

## Приклад 3

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main() {
```

```
int a=0,b=3,c;
```

```
c=b%2 || (a>=0)&& (++b/2*a)==0;
```

```
printf("a=%d, c=%d\n", a, c);
```

```
getch();
}
a=0, c=1
```

#### **Приклад 4**

```
#include <stdio.h>
#include <conio.h>
main() {
int a=1, b=0, c;
c=b%2 || (a>=0) && (++b*a)==0;
printf("c=%d\n", c);
getch();
}
c=0
```

#### **Приклад 5**

```
#include <stdio.h>
#include <conio.h>
main() {
int x=2, z, y=0;
z=(x==0) && (y=x) || (y>0);
printf("z=%d\n", z);
getch();
}
z=0
```

### **Відповіді на контрольні запитання**

#### **1) Пріоритети операцій:**

У мові C існує певний порядок виконання операцій, від найвищого пріоритету до найнижчого:

Дужки ()

Постфіксні оператори ++ і --

Префіксні оператори ++ і --

Оператори множення \*, ділення /, залишок від ділення %

Оператори додавання + і віднімання -

Оператори відношення <, <=, >, >=

Оператори рівності ==, !=

Логічні оператори I &&

Логічні оператори АБО ||

Оператор присвоєння =

Оператори побітового I &, АБО |, XOR ^

Оператори зсуву бітів <<, >>

#### **2) Таблиця істинності логічного I:**

Операція логічне І, виконується згідно таблиці істинності:

X	Y	$X \& Y$
0	0	0
0	1	0
1	0	0
1	1	1

3) Таблиця істинності логічного АБО:

Операція логічне АБО, виконується згідно таблиці істинності:

X	Y	$X \parallel Y$
0	0	0
0	1	1
1	0	1
1	1	1

4) Особливості виконання побітових операцій зсуву:

1. Операції зсуву вправо  $\gg$  та вліво  $\ll$  виконують зсув бітів вказаного числа на вказану кількість позицій.

2. При зсуві вправо знакове число може зберігати або втрачати свій знак в залежності від реалізації мови.

5) Таблиця істинності побітової операції XOR:

X	Y	$X \wedge Y$
0	0	0
0	1	1
1	0	1
1	1	0

**Висновок:** на цій лабораторній роботі я ознайомився і навчитися використовувати логічні та побітові операції під час програмування на мові C.