

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



**Лабораторна робота №3А**  
**з дисципліни «Програмування**  
**частина 2»**

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

**Прийняв:**  
Чайковський І.Б.

Львів 2024

**Тема роботи:** Логічні і бітові операції та вирази мови C.

**Мета роботи:** Дослідження властивостей операцій порівняння, логічних і бітових мови програмування C.

Теоретичні відомості

**Операції порівняння** – бінарні, причому обидва операнди повинні бути арифметичного типу, або вказівниками. Результат цілочисельний: 0 (хибність) або 1 (істинність). Тип результату int.

**Операція логічне АБО** виконується згідно таблиці істинності:

X	Y	X  Y
0	0	0
0	1	1
1	0	1
1	1	1

**Операція логічне І** виконується згідно таблиці істинності:

X	Y	X&&Y
0	0	0
0	1	0
1	0	0
1	1	1

### Пріоритети операцій

#### ОПЕРАЦІЯ

#### АСОЦІАТИВНІСТЬ

() [] -> .	ЗЛІВА НАПРАВО	→
! ~ ++ -- - (ТИП) * & sizeof	ЗЛІВА НАПРАВО	→
* / %	ЗЛІВА НАПРАВО	→
+ -	ЗЛІВА НАПРАВО	→
<< >>	ЗЛІВА НАПРАВО	→
< <= > >=	ЗЛІВА НАПРАВО	→
== !=	ЗЛІВА НАПРАВО	→
&	ЗЛІВА НАПРАВО	→
^	ЗЛІВА НАПРАВО	→
	ЗЛІВА НАПРАВО	→
&&	ЗЛІВА НАПРАВО	→
	ЗЛІВА НАПРАВО	→
?:	ЗПРАВА НАЛІВО	←
= += -= і.т.д.	ЗПРАВА НАЛІВО	←
, (кома)	ЗЛІВА НАПРАВО	→

### Приклад 1

```
#include <stdio.h>
void main(void)
{
    float var1, var2;
    printf("Введіть перше число (var1): ");
    scanf("%f", &var1);
    printf("Введіть друге число (var2): ");
    scanf("%f", &var2);
    printf("var1 > var2 дає %d\n", var1 > var2);
    printf("var1 < var2 дає %d\n", var1 < var2);

    printf("var1 == var2 дає %d\n", var1 == var2);
    printf("var1 >= var2 дає %d\n", var1 >= var2);
    printf("var1 <= var2 дає %d\n", var1 <= var2);
    printf("var1 != var2 дає %d\n", var1 != var2);
    printf("!var1 дає %d\n", !var1);
    printf("!var2 дає %d\n", !var2);
    printf("var1 || var2 дає %d\n", var1 || var2);
    printf("var1 && var2 дає %d\n", var1 && var2);
}
```

```
Введіть перше число (var1): 54
Введіть друге число (var2): 98
var1 > var2 дає 0
var1 < var2 дає 1
var1 == var2 дає 0
var1 >= var2 дає 0
var1 <= var2 дає 1
var1 != var2 дає 1
!var1 дає 0
!var2 дає 0
var1 || var2 дає 1
var1 && var2 дає 1
```

### Приклад 2

```
#include <stdio.h>
#define TRUE "ІСТИНА"
#define FALSE "ХИБНІСТЬ"
void main(void)
{
    float var1, var2;
    printf("Введіть перше число (var1): ");
    scanf("%f", &var1);
    printf("Введіть друге число (var2): ");
    scanf("%f", &var2);
```

```

printf("var1 > var2 це %s\n", var1 > var2 ? TRUE : FALSE);
printf("var1 < var2 це %s\n", var1 < var2 ? TRUE : FALSE);
printf("var1 == var2 це %s\n", var1 == var2 ? TRUE : FALSE);
printf("var1 >= var2 це %s\n", var1 >= var2 ? TRUE : FALSE);
printf("var1 <= var2 це %s\n", var1 <= var2 ? TRUE : FALSE);
printf("var1 != var2 це %s\n", var1 != var2 ? TRUE : FALSE);
printf("var1 || var2 це %s\n", var1 || var2 ? TRUE : FALSE);
printf("var1 && var2 це %s\n", var1 && var2 ? TRUE : FALSE);
printf("!var1 це %s\n", !var1 ? TRUE : FALSE);
printf("!var2 це %s\n", !var2 ? TRUE : FALSE);

```

```

Введіть перше число (var1): 67
Введіть друге число (var2): 39
var1 > var2 це ІСТИНА
var1 < var2 це ХИБНІСТЬ
var1 == var2 це ХИБНІСТЬ
var1 >= var2 це ІСТИНА
var1 <= var2 це ХИБНІСТЬ
var1 != var2 це ІСТИНА
var1 || var2 це ІСТИНА
var1 && var2 це ІСТИНА
!var1 це ХИБНІСТЬ
!var2 це ХИБНІСТЬ
}

```

### Приклад 3

```

#include <stdio.h>
int main() {
    int x, y, z;
    x = 2;
    y = 1;
    z = 0;
    x = x && y || z;
    printf("%d\n", x); // Виведе результат першого виразу
    printf("%d\n", x || !y && z); // Виведе результат другого виразу
}
1
1

```

### Приклад 4

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int a=0, b=3, c;

```

```

    c = b%2 || (a >= 0) && (++b/2*a)==0;
    printf("a=%d,c=%d\n",a,c); //a=0,c=1
    getch();
}
a=0, c=1

```

### Приклад 5

```

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

```

### Приклад 6

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int x=1, y=2, z;
    z=(x/2*7 <= 0) && (y<0) || (y%x==0);
    printf("z=%d\n", --z); //z=0
    getch();
}
z=0

```

### Приклад 7

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int x=1, z, b=0, y=2;
    z=(x++*y >= 0) || b++ || (x/y*3==0);
    printf("z=%d\n", z); //z=1
    getch();
}
z=1

```

### Приклад 8

```

#include <stdio.h>
#include <conio.h>
void main()

```

```

{
    int x=1,y=0,z=2;int a=0;
    z=((a=x++)*y==0 || a<0 && z);
    printf("z=%d\n",z); //z=1
    getch();
}
z=1

```

### Приклад 9

```

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

```

### Приклад 10

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int x=0,y=3,z;
    z=(++x>y || y-- && y>0);
    printf("z=%d\n",z); //z=1
    getch();
}
z=1

```

### Приклад 11

```

#include<stdio.h>
#include<conio.h>
void main()
{
    unsigned int x=2,y = 1, z=3,res;
    char chx = 0xAF;
    printf("%u\n",x&y|z); /*3*/
    x=y=z=2;
    printf("%u\n",x|y&z); /*2*/
    x=3; y=0; z=1;
    printf("x^y|~z=%hu\n",x^y|~z); /*65535*/
    printf("3|0^~1=%hu\n",x|y^~z); /*65535==11111111*/
    x=1;y=2;z=0;
    printf("1&2|0=%u\n",x&y|z); /*0*/
}

```

```

printf("~1^2&0=%hu\n",~x^y&z);/*65534==11111110*/
printf("2|0&1=%u\n",y|z&x);/*2*/
printf("2++&~0|~1=%hu\n",y++&~z|~x);/*65534==11111110*/
printf("~3|1&++0=%hu\n",~y|x&++z);/*65533==11111101*/
x = 0xAF;printf("%X\n",x>>4);/*A*/
chx<<=7;printf("0x=%X\n",chx);/*(FF)80 ==10000000*/
getch();
}

```

```

3
2
x^y|~z=65535
3|0^~1=65535
1&2|0=0
~1^2&0=65534
2|0&1=2
2++&~0|~1=65534
~3|1&++0=65533
A
0x=FFFFFFF80

```

## Приклад 12

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

```

int main() {
    unsigned char x = 255, y = 0177;
    printf("%hhu\n", x & y); // 127
    x = 40; // '(tm)' в ASCII коді відповідає 40
    y = 017;
    printf("%hhu\n", x & ~y); // 240
    y = 127;
    printf("%hhu\n", x & y); // 40
    y = 128;
    printf("%hhu\n", x | y); // 168
}

```

```

127
32
40
168

```

## Відповіді на контрольні запитання

1) Призначення операторів порівняння та тип результату:

Оператори порівняння використовуються для порівняння значень двох виразів. Результатом порівняння є логічне значення true або false. Тип результату - логічний (bool).

2) Особливість оператора "логічне дорівнює":

Оператор "логічне дорівнює" (==) перевіряє, чи рівні значення двох операндів. Він повертає true, якщо вони рівні, і false в протилежному випадку.

3) Чим відрізняються операнди в логічних операціях від операндів в операціях порівняння?

Операнди в логічних операціях - це значення, які можуть бути true або false. Операнди в операціях порівняння - це значення, які можуть бути будь-якого типу даних, і порівнюються за їхніми значеннями.

4) 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 <<, >>

5) Таблиця істинності логічного I:

Операція логічне I, виконується згідно таблиці істинності:

X	Y	X&&Y
0	0	0
0	1	0
1	0	0
1	1	1

6) 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

7) ) 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.

8) Порядок виконання бітових операцій I, АБО:

Операції I (AND) та АБО (OR) виконуються послідовно зліва направо.

9) Таблиця істинності бітової операції XOR(АБО):

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 дослідив властивостей операцій порівняння, логічних бітових мови програмування C.