

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи №3 а
з дисципліни «Програмування»

Підготував:
ст. групи АП-11
Чаус В. О.

Прийняла: Чайковський І. Б.

Тема:

Логічні і бітові операції та вирази мови C.

Мета:

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.

Операції рівності і нерівності відносять до цієї ж групи. Важливо правильно витримувати синтаксис знаку «логічне дорівнює» - ця операція не виконує присвоювання: вираз == вираз вираз != вираз

Результатом цих операцій є 0, якщо задане відношення хибне, і 1, якщо істинне. Тип результату int. Ці операції мають нижчий пріоритет, ніж операції попередньої групи, наприклад, у виразі $a < b == c < d$ спочатку здійснюються порівняння $a < b$ та $c < d$, результати кожного з них мають значення 0 або 1, після чого операція $==$ дає результат 0 або 1.

Ця операція повертає 1, якщо обидва операнди ненульові, та 0 в протилежному випадку. Операція гарантує обчислення зліва направо, а якщо лівий операнд є 0, правий не обчислюється. Операнди не повинні бути обов'язково однакового типу, але повинні мати один з основних типів. Результат завжди типу int.

Умовна тримісна операція (тернарна).

На відміну від унарних і бінарних операцій умовна тернарна операція використовується з трьома операндами. В зображенні умовної операції використовуються два символи '?' і ':' і три вирази: вираз1 ? вираз2 : вираз3.

Першим обчислюється виразу1. Якщо воно істинне, тобто не дорівнює нулю, то обчислюється значення виразу2, яке стає результатом. Якщо при обчисленні виразу 1 отримується 0 (нуль), то в якості результату приймається значення виразу3. Приклад:

$x < 0 ? -x : x$;

Вираз повертає абсолютну величину змінної x.

Завдання:

1. Здійснити виконання програми порівняння двох чисел :

```
1 #include <stdio.h>
2
3 void main(void) {
4     float var1, var2;
5     printf("Введіть перше число (var1): ");
6     scanf("%f", &var1);
7     printf("Введіть друге число (var2): ");
8     scanf("%f", &var2);
9
10    printf("var1 > var2 дає %d\n", var1 > var2);
11    printf("var1 < var2 дає %d\n", var1 < var2);
12    printf("var1 == var2 дає %d\n", var1 == var2);
13    printf("var1 >= var2 дає %d\n", var1 >= var2);
14    printf("var1 <= var2 дає %d\n", var1 <= var2);
15    printf("var1 != var2 дає %d\n", var1 != var2);
16    printf("!var1 дає %d\n", !var1);
17    printf("!var2 дає %d\n", !var2);
18    printf("var1 || var2 дає %d\n", var1 || var2);
19    printf("var1 && var2 дає %d\n", var1 && var2);
20 }
21
```

Введіть перше число (var1): 5
Введіть друге число (var2): 1
var1 > var2 дає 1
var1 < var2 дає 0
var1 == var2 дає 0
var1 >= var2 дає 1
var1 <= var2 дає 0
var1 != var2 дає 1
!var1 дає 0
!var2 дає 0
var1 || var2 дає 1
var1 && var2 дає 1

2. Здійснити модифікацію та виконання програми згідно вірця, показаного нижче..

```

1  #include <stdio.h>
2  #define TRUE "ІСТИНА"
3  #define FALSE "ХИБНІСТЬ"
4
5  void main(void) {
6      float var1, var2;
7      printf("Введіть перше число (var1): ");
8      scanf("%f", &var1);
9      printf("Введіть друге число (var2): ");
10     scanf("%f", &var2);
11     printf("var1 > var2 це %s\n", var1 > var2 ? TRUE : FALSE);
12     printf("var1 < var2 це %s\n", var1 < var2 ? TRUE : FALSE);
13     printf("var1 == var2 це %s\n", var1 == var2 ? TRUE : FALSE);
14     printf("var1 >= var2 це %s\n", var1 >= var2 ? TRUE : FALSE);
15     printf("var1 <= var2 це %s\n", var1 <= var2 ? TRUE : FALSE);
16     printf("var1 != var2 це %s\n", var1 != var2 ? TRUE : FALSE);
17     printf("var1 || var2 це %s\n", var1 || var2 ? TRUE : FALSE);
18     printf("var1 && var2 це %s\n", var1 && var2 ? TRUE : FALSE);
19     printf("!var1 це %s\n", !var1 ? TRUE : FALSE);
20     printf("!var2 це %s\n", !var2 ? TRUE : FALSE);
21 }
22

```

input

```

Введіть перше число (var1): 10
Введіть друге число (var2): 10
var1 > var2 це ХИБНІСТЬ
var1 < var2 це ХИБНІСТЬ
var1 == var2 це ІСТИНА
var1 >= var2 це ІСТИНА
var1 <= var2 це ІСТИНА
var1 != var2 це ХИБНІСТЬ
var1 || var2 це ІСТИНА
var1 && var2 це ІСТИНА
!var1 це ХИБНІСТЬ
!var2 це ХИБНІСТЬ

```

3. Створити програму для виконання прикладу:

```
1  #include <stdio.h>
2
3  void main(void) {
4      int x, y, z;
5      x = 2;
6      y = 1;
7      z = 0;
8      x = x && y || z;
9      printf("%d\n", x);
10     printf("%d\n", x || !y && z);
11 }
12
```

```
1
1
1  #include <stdio.h>
2  #include <conio.h>
3
4  void main() {
5      int a = 0, b = 3, c;
6      c = b % 2 || (a >= 0) && (++b / 2 * a) == 0;
7      printf("a=%d, c=%d\n", a, c);
8      getch();
9  }
10
```

a=0, c=1

```

1  #include <stdio.h>
2  #include <conio.h>
3
4  void main() {
5      int x = 1, z, b = 0, y = 2;
6      z = (x++ * y >= 0) || b++ || (x / y * 3 == 0);
7      printf("z=%d\n", z);
8      getch();
9  }
10

```

z=1

```

1  #include <stdio.h>
2  #include <conio.h>
3
4  void main() {
5      int x = 1, y = 0, z = 2;
6      int a = 0;
7      z = ((a = x++) * y == 0 || a < 0 && z);
8      printf("z=%d\n", z);
9      getch();
10 }
11

```

z=1

...Program finished with exit code 255
Press ENTER to exit console.

z=0

```
1 #include <stdio.h>
2 #include <conio.h>
3
4 void main() {
5     int x = 2, z, y = 0;
6     z = (x == 0) && (y = x) || (y > 0);
7     printf("z=%d\n", z);
8     getch();
9 }
10
```



z=0

main.c

```
1 #include <stdio.h>
2 #include <conio.h>
3
4 void main() {
5     int x = 0, y = 3, z;
6     z = (++x > y || y-- && y > 0);
7     printf("z=%d\n", z);
8     getch();
9 }
10
```

z=1

...Program finished with exit code 255
Press ENTER to exit console.


```

1  #include <stdio.h>
2  #include <conio.h>
3
4  void main() {
5      unsigned int x = 2, y = 1, z = 3, res;
6      char chx = 0xAF;
7
8      printf("%u\n", x & y | z);
9      x = y = z = 2;
10     printf("%u\n", x | y & z);
11     x = 3; y = 0; z = 1;
12     printf("x^y|~z=%u\n", x ^ y | ~z);
13     printf("3|0^~1=%u\n", 3 | 0 ^ ~1);
14     x = 1; y = 2; z = 0;
15     printf("1&2|0=%u\n", x & y | z);
16     printf("~1^2&0=%u\n", ~x ^ y & z);
17     printf("2|0&1=%u\n", y | z & x);
18     printf("2++&~0|~1=%u\n", ++y & ~z | ~x);
19     printf("~3|1&++0=%u\n", ~y | x & ++z);
20     x = 0xAF;
21     printf("%X\n", x >> 4);
22     chx <<= 7;
23     printf("0x=%X\n", chx);
24     getch();
25 }
26

```

```

3
2
x^y|~z=4294967295
3|0^~1=4294967295
1&2|0=0
~1^2&0=4294967294
2|0&1=2
2++&~0|~1=4294967295
~3|1&++0=4294967293
A
0x=FFFFFFF80

...Program finished with exit code 255
Press ENTER to exit console.

```

```
1 #include <stdio.h>
2 #include <conio.h>
3
4 void main() {
5     char x = 255, y = 0177;
6     printf("%u\n", (unsigned int)x & (unsigned int)y);
7     x = '\0';
8     y = 017;
9     printf("%u\n", (unsigned int)x & ~(unsigned int)y);
10    y = 127;
11    printf("%u\n", (unsigned int)x & (unsigned int)y);
12    y = 128;
13    printf("%u\n", (unsigned int)x | (unsigned int)y);
14 }
15
```

127
0
0
4294967168

...Program finished with exit code 11
Press ENTER to exit console.

Контрольні запитання:

Ось відповіді на ваші контрольні запитання:

1. Призначення операторів порівняння та тип результату :
 - Оператори порівняння використовуються для порівняння значень. Тип результату - логічне значення (true або false), що вказує на те, чи вірне порівняння.
2. Особливість оператора "логічне дорівнює" :
 - Оператор "логічне дорівнює" (==) порівнює значення обох операндів і їх типи даних. Це означає, що не тільки значення повинні бути однакові, але й типи даних повинні бути однакові.
3. Відрізняються операнди в логічних операціях від операндів в операціях порівняння:
 - Операнди в логічних операціях це логічні значення (true або false), тоді як операнди в операціях порівняння - це значення, що порівнюються.
4. Пріоритети операцій :

- Порядок виконання операцій може залежати від пріоритету. Наприклад, зазвичай арифметичні операції мають вищий пріоритет, ніж логічні.

5. Таблиця істинності логічного І :

A	B	A І B
0	0	0
0	1	0
1	0	0
1	1	1

6. Таблиця істинності логічного АБО :

A	B	A АБО B
0	0	0
0	1	1
1	0	1
1	1	1

7. Особливості виконання бітових операцій зсуву :

- При зсуві вправо операнди зсуваються на вказану кількість позицій вправо, а при зсуві вліво - на вказану кількість позицій вліво. При цьому звільнені позиції заповнюються нулями.

8. Порядок виконання бітових операцій І, АБО :

- Порядок виконання бітових операцій І (AND) та АБО (OR) зазвичай відбувається зліва направо.

9. Таблиця істинності бітової операції XOR :

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0