

Міністерство освіти і науки України  
Інститут телекомунікацій, радіоелектроніки та електронної техніки  
**Звіт до лабораторної роботи № 3А**

**Тема: «Логічні і бітові операції та вирази мови С»**  
з дисципліни «Програмування частина 2»  
Варіант № 6

виконав студент групи АП-11  
Головацький Назар  
перевірив доцент кафедри ТК  
Чайковський І.Б

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

## ЗАВДАННЯ

1.

```
#include <stdio.h>

int 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);

    return 0;
}
```

## Результат:

Введіть перше число (var1): 1

Введіть друге число (var2): 9

`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

2.

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

```
#define TRUE "ІСТИНА"
```

```
#define FALSE "ХИБНІСТЬ"
```

```
int 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);
```


```
    return 0;
```

```
}
```

## Результат:

Введіть перше число (var1): 3

Введіть друге число (var2): 6

var1 > var2 це ХИБНІСТ 

var1 < var2 це ІСТИНА

var1 == var2 це ХИБНІСТЬ

var1 >= var2 це ХИБНІСТЬ

var1 <= var2 це ІСТИНА

var1 != var2 це ІСТИНА

var1 || var2 це ІСТИНА

var1 && var2 це ІСТИНА

!var1 це ХИБНІСТЬ

!var2 це ХИБНІСТЬ

## 3.

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

```
int main() {  
    int x, y, z;  
    x = 2;  
    y = 1;  
    z = 0;  
    x = x && y || z;  
    printf("%d\n", x);  
  
    printf("%d\n", x || !y && z);  
    return 0;  
}
```

## Результат:

1

1

4.

1)

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

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

```
void main() { // Початок визначення функції main без параметрів, що повертає  
значення int.
```

```
    int a = 0, b = 3, c; // Оголошення та ініціалізація змінних типу int: a зі  
значенням 0, b зі значенням 3, c - без ініціалізації.
```

```
    c = b % 2 || (a >= 0) && (++b / 2 * a) == 0; // Присвоєння змінній c результату  
логічного виразу.
```

```
    printf("a=%d,c=%d\n", a, c); // Виведення значень змінних a та c на екран.
```

```
    /*a=0,c=1*/ // Коментар до виводу, що пояснює значення виведених змінних.
```

```
    getch(); // Очікування натискання клавіші користувачем перед завершенням  
програми.
```

```
}
```

2)

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

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

```
void main() { // Початок визначення функції main без параметрів, що повертає  
значення int.
```

```
    int a = 1, b = 0, c; // Оголошення та ініціалізація змінних типу int: a зі  
значенням 1, b зі значенням 0, c - без ініціалізації.
```

```
    c = b * 2 || (a >= 0) && (++b * a) == 0; // Присвоєння змінній c результату  
логічного виразу.
```

```
    printf("c=%d\n", c); // Виведення значення змінної c на екран.
```

```
    /*c=0*/ // Коментар до виводу, що пояснює значення виведеної змінної.
```

```
    getch(); // Очікування натискання клавіші користувачем перед завершенням  
програми.
```

```
}
```

3)

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

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

```
void main() { // Початок визначення функції main без параметрів, що повертає  
значення int.
```

```
    int x = 1, y = 2, z; // Оголошення та ініціалізація змінних типу int: x зі  
значенням 1, y зі значенням 2, z - без ініціалізації.
```

```
    z = (x / 2 * 7 <= 0) && (y < 0) || (y % x == 0); // Присвоєння змінній z результату  
логічного виразу.
```

```
    printf("z=%d\n", --z); // Виведення значення змінної z на екран після зменшення  
на одиницю.
```

```
    /*z=0*/ // Коментар до виводу, що пояснює значення виведеної змінної.
```

```
    getch(); // Очікування натискання клавіші користувачем перед завершенням  
програми.
```

```
}
```

4)

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

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

```
void main() { // Початок визначення функції main без параметрів, що повертає  
значення int.
```

```
    int x = 1, z, b = 0, y = 2; // Оголошення та ініціалізація змінних типу int: x зі  
значенням 1, z - без ініціалізації, b зі значенням 0, y зі значенням 2.
```

```
    z = (x++ * y >= 0) || b++ || (x / y * 3 == 0); // Присвоєння змінній z результату  
логічного виразу.
```

```
    printf("z=%d\n", z); // Виведення значення змінної z на екран.
```

```
    /*z=1*/ // Коментар до виводу, що пояснює значення виведеної змінної.
```

```
    getch(); // Очікування натискання клавіші користувачем перед завершенням  
програми.
```

```
}
```

5)

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

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

```
void main() { // Початок визначення функції main без параметрів, що повертає  
значення int.
```

```
    int x = 1, y = 0, z = 2; // Оголошення та ініціалізація змінних типу int: x зі  
значенням 1, y зі значенням 0, z зі значенням 2.
```

```
    int a = 0; // Ініціалізація змінної a значенням 0.
```

```
    z = ((a = x++) * y == 0 || a < 0 && z); // Присвоєння змінній z результату  
логічного виразу.
```

```
    printf("z=%d\n", z); // Виведення значення змінної z на екран.
```

```
    /*z=1*/ // Коментар до виводу, що пояснює значення виведеної змінної.
```

```
    getch(); // Очікування натискання клавіші користувачем перед завершенням  
програми.
```



```
}
```

6)

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

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

```
void main() { // Початок визначення функції main без параметрів, що повертає  
значення int.
```

```
    int x = 2, z, y = 0; // Оголошення та ініціалізація змінних типу int: x зі  
значенням 2, z - без ініціалізації, y зі значенням 0.
```

```
    z = (x == 0) && (y = x) || (y > 0); // Присвоєння змінній z результату логічного  
виразу.
```

```
    printf("z=%d\n", z); // Виведення значення змінної z на екран.
```

```
    /*z=0*/ // Коментар до виводу, що пояснює значення виведеної змінної.
```

```
    getch(); // Очікування натискання клавіші користувачем перед завершенням  
програми.
```

```
}
```

7)

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

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

```
void main() { // Початок визначення функції main без параметрів, що повертає  
значення int.
```

```
    int x = 0, y = 3, z; // Оголошення та ініціалізація змінних типу int: x зі  
значенням 0, y зі значенням 3, z - без ініціалізації.
```

```
    z = (++x > y || y-- && y > 0); // Присвоєння змінній z результату логічного  
виразу.
```

```
    printf("z=%d\n", z); // Виведення значення змінної z на екран.
```

```
    /*z=1*/ // Коментар до виводу, що пояснює значення виведеної змінної.
```

```
    getch(); // Очікування натискання клавіші користувачем перед завершенням  
програми.
```

```
}
```

5.

```
#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);
```

```
    x = y = z = 2;
```

```
    printf("%u\n", x | y & z);
```

```
    x = 3;
```

```
    y = 0;
```

```
    z = 1;
```

```
    printf("x^y|~z=%u\n", x ^ y | ~z);
```

```
    printf("3|0^~1=%u\n", x | y ^ ~z);
```

```
    x = 1;
```

```
    y = 2;
```

```
    z = 0;
```

```
    printf("1&2|0=%u\n", x & y | z);
```

```
    printf("~1^2&0=%u\n", ~x ^ y & z);
```

```
    printf("2|0&1=%u\n", y | z & x);
```

```
    printf("2++&~0|~1=%u\n", y++ & ~z | ~x);
```

```
    printf("~3|1&++0=%u\n", ~y | x & ++z);
```

```
    x = 0xAF;
```

```
    printf("%X\n", x >> 4);
```

```
    chx <<= 7;
```

```
    printf("0x=%X\n", chx);
```

```
    getch();
```

```
}
```

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

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

```
void main() {  
    char x = 255, y = 0177;  
    printf("%u\n", x & y);  
    x = '(';  
    y = 017;  
    printf("%u\n", x & ~y);  
    y = 127;  
    printf("%u\n", x & y);  
    y = 128;  
    printf("%u\n", x | y);  
}
```

**Висновок:** Я вивчив логічні і бітові операції та вирази мови C