

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
Кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи № 9
«Програмування, частина 2»

Підготував:
ст. гр. ІХ-11
Диркавець Максим
Перевірив:
Асистент каф РЕПС
Чайковський
І.Б.

Тема: Логічні та побітові операції у мові C

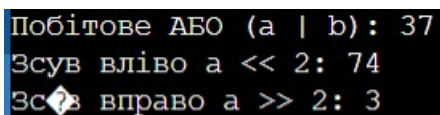
Мета роботи: навчитися використовувати логічні та побітові операції під час програмування на мові C.

Хід роботи

1. Написати програму на мові C, яка здійснює такі побітові операції як побітове І, побітове АБО, зсув вліво на 2, зсув вправо на 2. Дані операції застосувати до змінних: a= 017, b=036 (змінні представлені у вісімковій системі числення). Операцію зсуву застосувати тільки до змінної a. Скрін коду програми та результати її виконання представити у звіті.

```
#include <stdio.h>

int main() {
    int a = 017;
    int b = 036;
    int and = a & b;
    int or = a | b;
    int left = a << 2;
    int right = a >> 2;
    printf("Початкові значення: a = %o, b = %o\n", a, b);
    printf("Побітове І (a & b): %o\n", and);
    printf("Побітове АБО (a | b): %o\n", or);
    printf("Зсув вліво a << 2: %o\n", left);
    printf("Зсув вправо a >> 2: %o\n", right);
    return 0;
}
```



```
Побітове АБО (a | b): 37
Зсув вліво a << 2: 74
Зсув вправо a >> 2: 3
```

2. Здійснити вручну виконання операцій з пункту 1. Для цього здійснити

переведення значень змінних $a=017$, $b=036$ з вісімкової у двійкову систему числення та виконати необхідні операції згідно п.1. Отримані результати представити у звіті, та порівняти їх з результатами програми з пункту 1.

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

$$a = 017_8 = 000\ 001\ 111_2$$

$$b = 036_8 = 011\ 110_2$$

І ($a \& b$)

$$000\ 001\ 111 \& 000\ 011\ 110 = 000\ 001\ 110$$

АБО ($a | b$)

$$000\ 001\ 111 | 000\ 011\ 110 = 000\ 011\ 111$$

Зсув вліво на 2; $a \ll 2$:

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

Зсув вправо на 2; $a \gg 2$:

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

Результати для a та b

$$a \& b = 000\ 001\ 110_2 = 016_8$$

$$a | b = 000\ 011\ 111_2 = 037_8$$

$$a \ll 2 = 000\ 111\ 100_2 = 074_8$$

$$a \gg 2 = 000\ 000\ 011_2 = 003_8$$

3. Здійснити виконання прикладів, представлених нижче. Представити скріни

коду та результати їх виконання у звіті. Пояснити отримані результати.

Приклад 1

```
#include <stdio.h>
#include <conio.h>
int main() {
    int a = 0, b = 3, c;
    c = (b % 2) || ((a >= 0) && ((++b / 2) * a) == 0);
    printf("a = %d, b = %d, c = %d\n", a, b, c);
    getch();
    return 0;
}
```

```
a = 0, b = 3, c = 1
```

Пояснення:

Значення змінних після виконання виразу:

$a = 0$ (значення не змінилося)

$b = 4$ (значення змінилося через операцію $++b$)

$c = 1$ (результат обчислення логічного виразу)

Цей результат показує, що початкове значення змінної **a** залишилося незмінним, змінна **b** була збільшена на 1 через префіксну інкрементацію $++b$, і змінна **c** отримала значення **1**, оскільки результат логічного виразу був істинним.

Приклад 2

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

```
    getch();  
    return 0;  
}
```

c = 0

Пояснення:

Значення змінних після виконання виразу:

a = 1 (значення не змінилося)

b = 1 (значення змінилося через операцію ++**b**)

c = 0 (результат обчислення логічного виразу)

Цей результат показує, що початкове значення змінної **a** залишилося незмінним, змінна **b** була збільшена на 1 через префіксну інкрементацію ++**b**, і змінна **c** отримала значення **0**, оскільки результат логічного виразу був хибним.

Приклад 3

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

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

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

z = 0

Пояснення:

Значення змінних після виконання виразу:

x = 2 (значення не змінилося)

y = 0 (значення не змінилося, оскільки (**y** = **x**) не виконувався)

$z = 0$ (результат обчислення виразу)

Цей результат показує, що початкові значення змінних x і y залишилися незмінними, оскільки присвоєння $y = x$ не виконалося через те, що умова ($x == 0$) була хибною. Логічні операції в кінцевому результаті дали значення 0 для змінної z , що підтверджує правильність обчислення логічних виразів у програмі.

4. Оформити звіт.

Висновок: В ході даної роботи ми навчилися використовувати логічні та побітові операції в мові програмування C. Розглянули такі операції як побітове І, побітове АБО, зсув вліво та вправо. Провели практичні вправи з використанням цих операцій на конкретних прикладах, як в програмі, так і вручну.

Результати виконання операцій в програмі і вручну відповідали один одному, що свідчить про правильність реалізації цих операцій у програмі. Це дає нам зрозуміти, що ми правильно розуміємо принципи роботи цих операцій і можемо успішно використовувати їх у подальших програмах.