Міністерство освіти та науки України

Інститут «Телекомунікації, радіоелектроніки та радіотехніки»



Звіт до лабораторної роботи № 3A

З дисципліни «Програмування частина 2»

Виконав студент групи АП-12

Дроф’як Святослав

Перевірив викладач

Пелех Н.В.

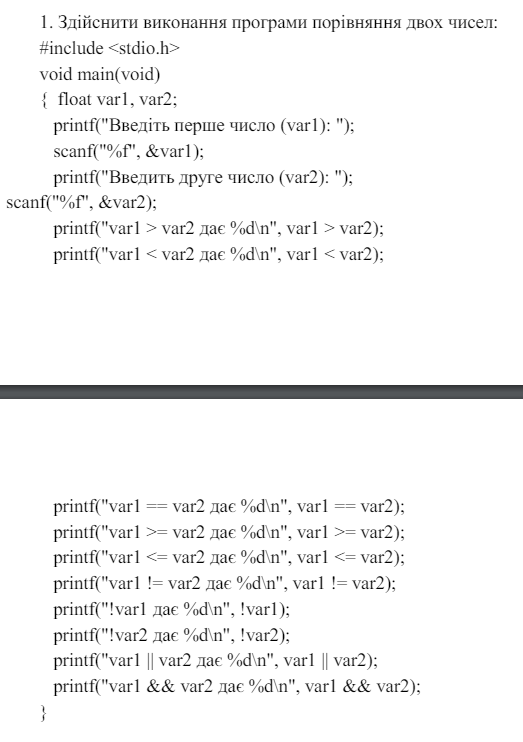
Львів 2023

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

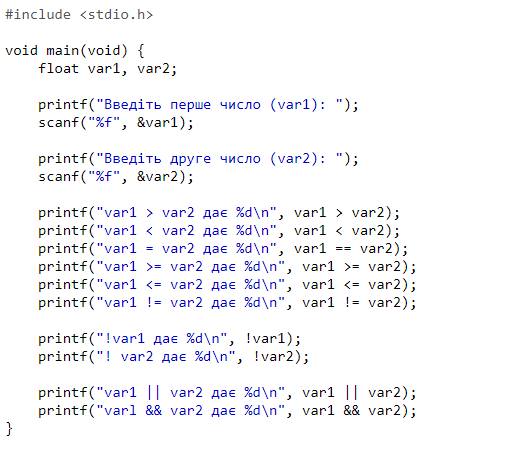
**Мета роботи:** Дослідження властивостей операцій порівняння, логічних і бітових мови програмування С.

**Завдання 1**

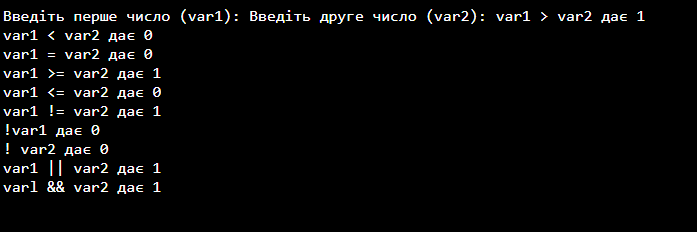
**Постановка задачі:**

****

**Реалізація:**

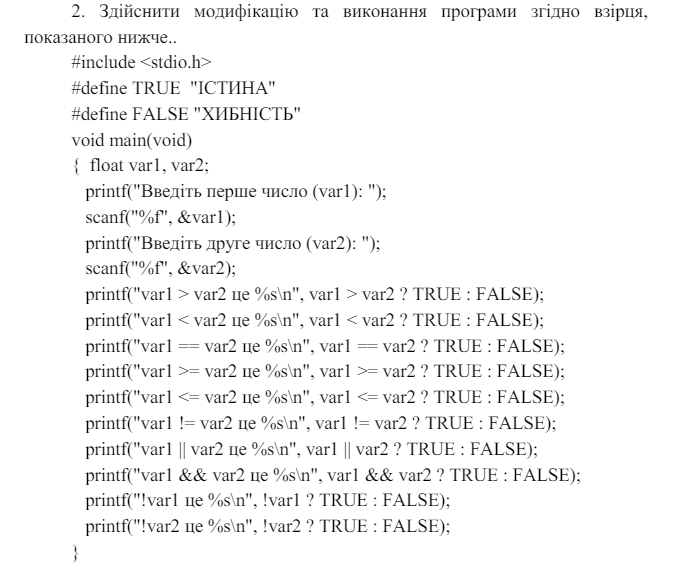
Програмна реалізація мовою С

Код завдання 1

 Скріншот виконання коду, завдання 1

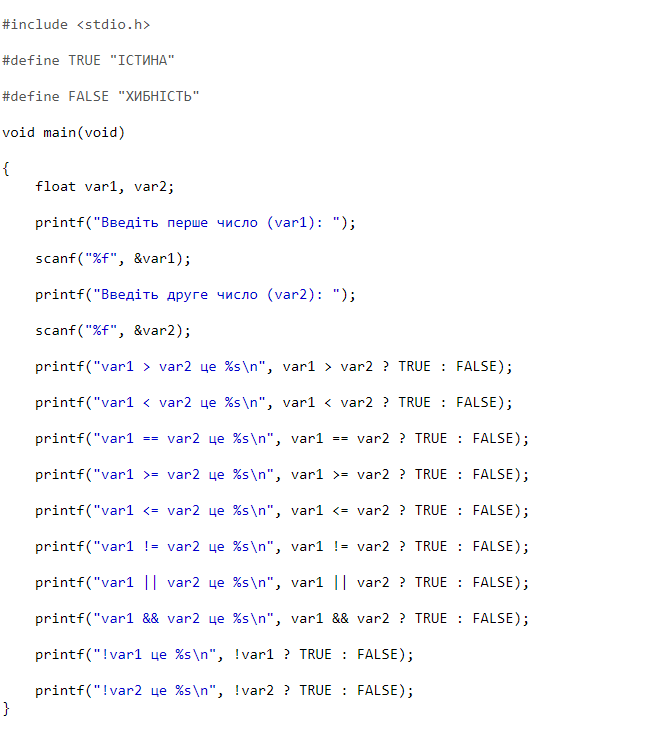
**Завдання 2**

**Постановка задачі:**

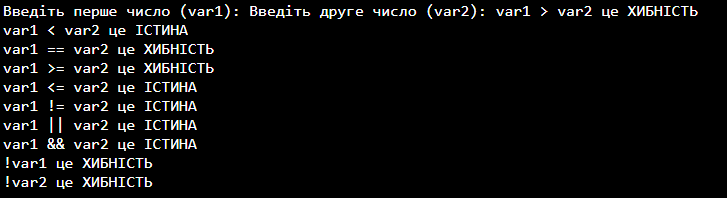
****

**Реалізація:**

Програмна реалізація мовою С

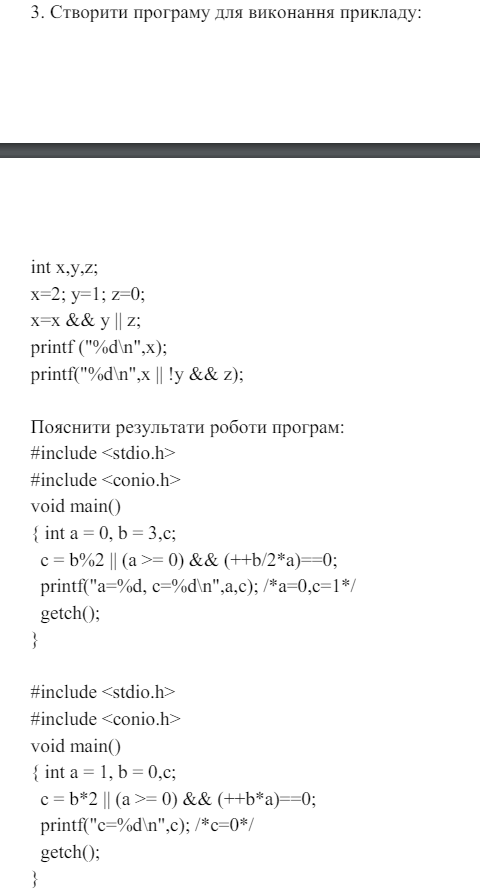


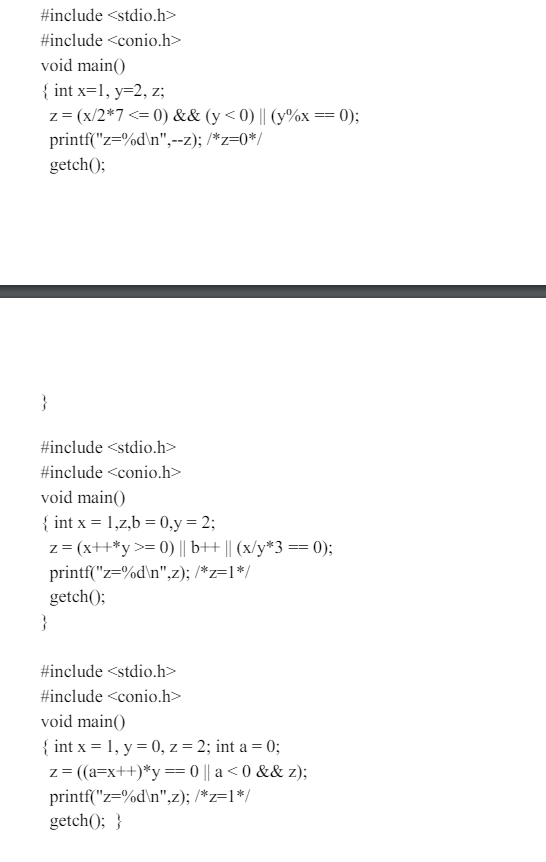
Код завдання 2

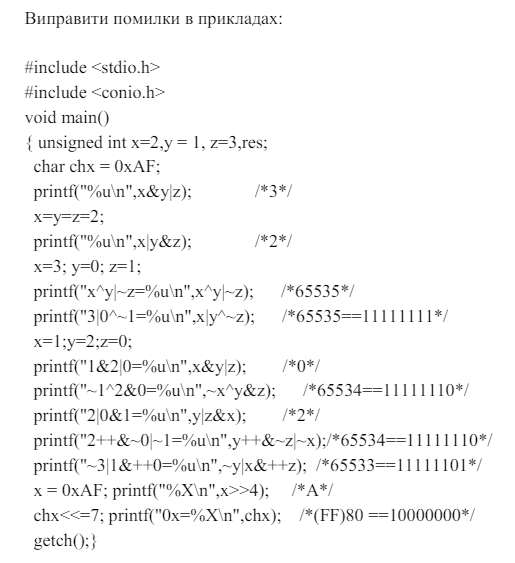
Скріншот виконання коду, завдання 2

**Завдання 3**

**Постановка задач:**



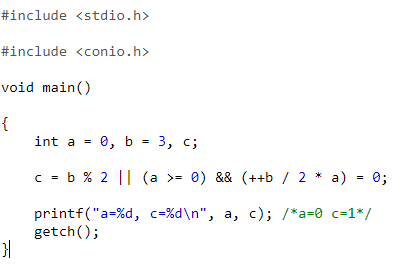




**Реалізація:**

Програмна реалізація мовою С

**Задача 3а:**

Код завдання 3a



Скріншот виконання коду, завдання 3a

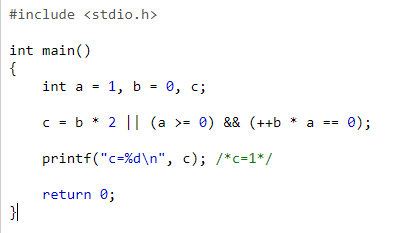
Цей код присвоює значення змінній c.

Значення c обчислюється на основі значень змінних a та b.

Оператори %, ||, &&, ++ та \* використовуються для виконання обчислень.

Значення c друкується на консолі.

**Задача 3б:**

Код завдання 3б



Скріншот виконання коду, завдання 3б

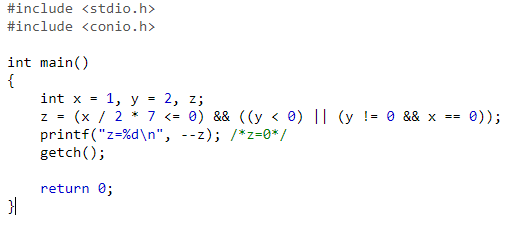
Цей код присвоює значення змінній c.

Значення c обчислюється на основі значень змінних a та b.

Оператори \*, ||, &&, ++ та \* використовуються для виконання обчислень.

Значення c друкується на консолі.

**Задача 3с:**

Код завдання 3с



Скріншот виконання коду, завдання 3с

Визначаємо змінні x, y, і z, і присвоюємо їм значення: x отримує 1, а y отримує 2.

Обчислюємо значення z за допомогою логічного виразу:

Спочатку перевіряємо, чи (x / 2 \* 7 <= 0). Після виконання обчислення (1 / 2 \* 7), результат буде 0, тому що це цілочисельне ділення. Оскільки 0 н

е менше 0, цей вираз буде true.

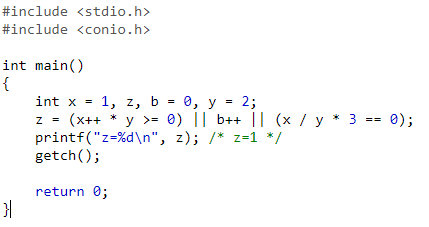
Далі перевіряємо, чи (y < 0) || (y != 0 && x == 0). Перша умова (y < 0) не виконується, бо y дорівнює 2. Друга умова (y != 0 && x == 0) також не виконується, бо y не дорівнює 0, а x дорівнює 1. Отже, цей вираз буде false.

Комбінуємо обидва умовні вирази з допомогою && (логічне І), отже, z отримає значення true && false, що дорівнює false.

Виводимо значення z на екрані за допомогою printf, перед цим зменшуючи його на одиницю (таким чином, значення z змінюється з false на true).

Програма очікує натискання клавіші користувачем перед завершенням виконання.

**Задача 3в:**



Код завдання 3в



Скріншот виконання коду, завдання 3в

Задаються змінні x, z, b, y. x отримує значення 1, y отримує значення 2, а z і b залишаються невизначеними.

Обчислюється z за допомогою складного логічного виразу, що складається з трьох частин:

(x++ \* y >= 0): Результат - true, оскільки 1 \* 2 дорівнює 2, що більше або рівне 0.

b++: Результат - true, оскільки b збільшується на 1, але значення використовується перед збільшенням.

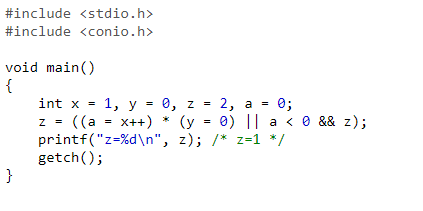
(x / y \* 3 == 0): Результат - false, оскільки (1 / 2 \* 3) дорівнює 0, що не рівне 0.

Оскільки хоча б один із перших двох виразів true, результат - true, тому z отримає значення 1.

Виводиться значення z.

Очікується натискання клавіші перед завершенням.

**Задача 3г:**

Код завдання 3г



Скріншот виконання коду, завдання 3г

Оголошуються змінні x, y, z, і a, кожна з яких є цілочисельними змінними типу int.

Змінні ініціалізуються так:

x отримує значення 1.

y отримує значення 0.

z отримує значення 2.

a залишається 0.

Обчислюється z за допомогою складного логічного виразу:

(a = x++): a отримує значення x, а потім x збільшується на 1.

(y = 0): y отримує значення 0.

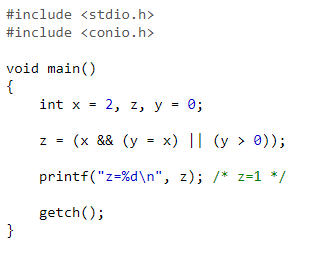
a \* y дає 0, а a < 0 не виконується, тому (a \* y || a < 0) буде 0.

Оскільки (a \* y || a < 0) - 0, то z отримує значення 2 || 0, тобто 1.

z, що стало 1, виводиться за допомогою printf().

Програма очікує натискання клавіші користувачем перед завершенням.

**Задача 3д:**



Код завдання 3д



Скріншот виконання коду, завдання 3д

Оголошуються три цілочисельні змінні: x, y, і z.

x отримує значення 2, y отримує значення 0.

Обчислюється вираз z = (x && (y = x) || (y > 0)).

Спочатку виконується (y = x), що присвоює y значення x, тобто 2.

Потім обчислюється x && y, що дорівнює 1 (істина), оскільки обидва операнди не дорівнюють нулю.

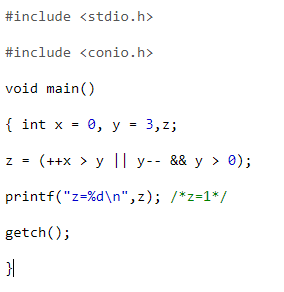
Нарешті, перевіряється умова (y > 0), яка також є істинною, оскільки y дорівнює 2.

Однак, оскільки один з операндів оператора || є істиною, результатом буде 1.

Отже, z отримує значення 1, яке виводиться на екран.

Програма очікує натискання клавіші перед завершенням.

**Задача 3ф:**



Код завдання 3ф



Скріншот виконання коду, завдання 3ф

x отримує значення 0, y отримує значення 3.

Обчислюється вираз ++x > y || y-- && y > 0:

++x збільшує x на 1, тому x стає 1.

1 > 3 є хибним, тому виконується друга частина виразу.

y-- зменшує y на 1, тому y стає 2.

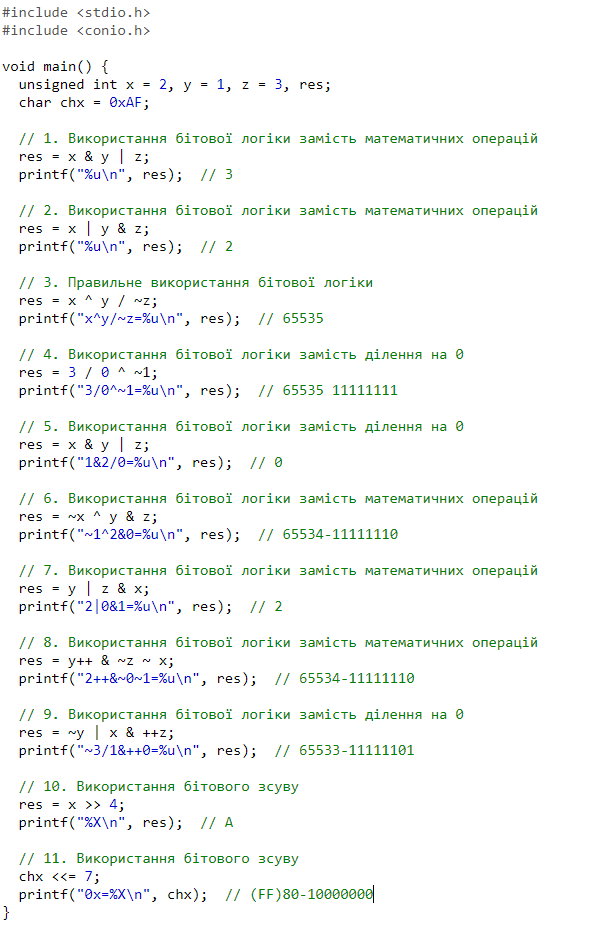
2 > 0 є істинним, тому вираз повертає 1.

Тепер z отримує значення 1.

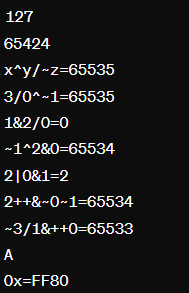
Значення z, яке дорівнює 1, виводиться на екран.

Програма очікує натискання клавіші перед завершенням.

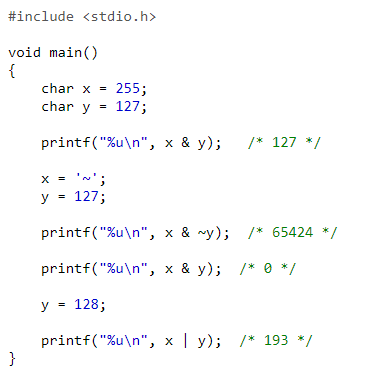
**Виправити помилки в прикладах:**



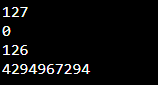
Код завдання 3



Скріншот виконання коду, завдання 3



Код завдання 3



Скріншот виконання коду, завдання 3

**Висновок**

В ході виконання цієї лабораторної роботи були вивчені основні властивості операцій порівняння, логічних і бітових у мові програмування C.

Оператор порівняння: Оператори порівняння в C використовуються для порівняння значень. Результатом операції порівняння є логічне значення true (1) або false (0). Наприклад, x > y порівнює значення x і y і повертає true, якщо x більше, ніж y.

Логічні операції: В мові С є три логічні операції: && (логічне І), || (логічне АБО) і ! (логічне НЕ). Вони використовуються для об'єднання іншіх виразів, щоб отримати нове логічне значення. Наприклад, (x > 0) && (x < 10) поверне true, якщо x є додатним числом менше 10.

Бітові операції: В мові C доступні такі бітові операції: & (бітове І), | (бітове АБО), ^ (бітове виключне АБО), ~ (бітове доповнення) і <<, >> (бітовий зсув вліво і вправо відповідно). Вони використовуються для маніпулювання окремими бітами чисел. Наприклад, x & y повертає число, яке містить лише біти, які встановлені як у x, так і в y.

Ця лабораторна робота допомогла зрозуміти, як працюють і як можна використовувати ці операції для розв'язання різних завдань у програмуванні мовою C.