МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ ТА РАДІОЕЛЕКТРОНІКИ



**Лабораторна робота №7**

**Виконав :**

Ст. гр АП-11

Білоніжко Віталій

**Прийняв:**

Чайковський І. Б.

**Львів 2024**

**Тема:**

Арифметичні операції та вирази мови C.

**Мета:**

ознайомитися з синтаксисом арифметичних операцій, їх пріоритетом застосувань, навчитися їх використовувати для обчислень математичних виразів.

**Теоретичні відомості:**

Мова С була розроблена в процесі створення операційної системи UNIX, тому можна зрозуміти, які принципові можливості в ній реалізовані: це максимальна гнучкість при діалоговому режимі роботи комп’ютера, представлення повідомлень системи і користувача в максимально простій і зрозумілій формі і, водночас, спроможність вибору адекватної реакції в найскладніших ситуаціях. Мова С поєднує в собі можливості прямої адресації і побітових операцій, як в Ассемблері, з використанням великої кількості (декілька сотень) функцій найвищого рівня. При використанні бібліотеки графічних функцій мова С отримала практично необмежені можливості для розробки діалогових програмних засобів.

**Хід роботи:**

1. Ознайомитися з теоретичними відомостями.

2. Здійснити виконання усіх прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати їх виконання у звіті.

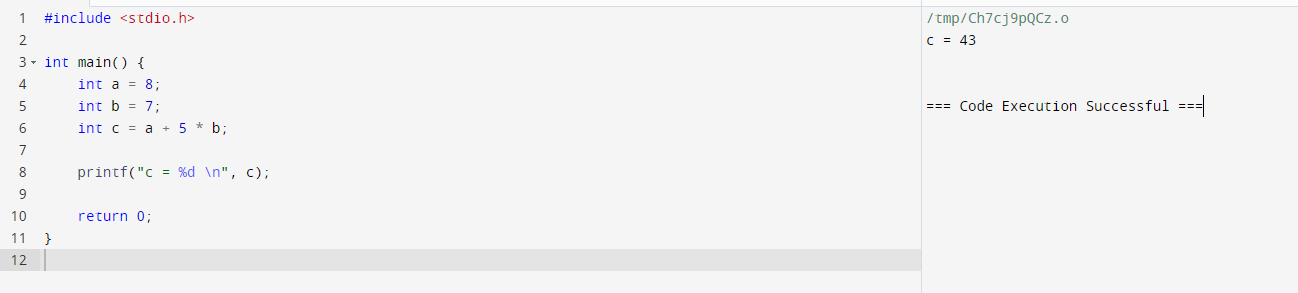
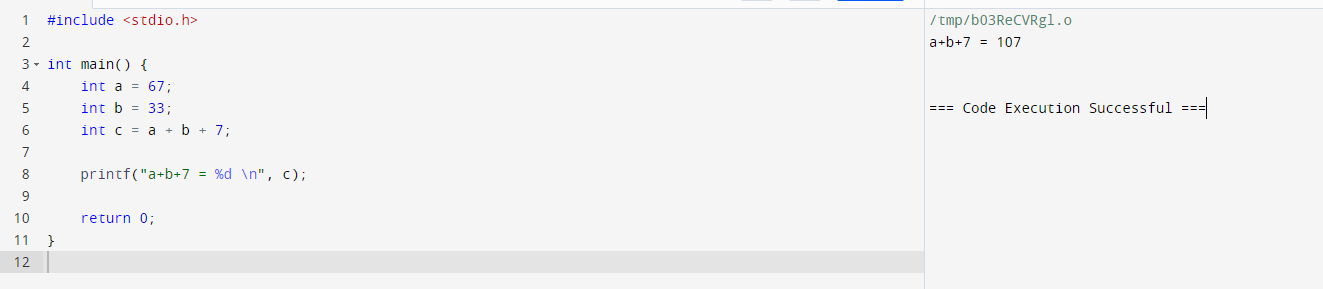
3. Набрати текст нижченаведеної програми, виправити усі синтаксичні помилки, здійснити її компіляцію. Скрін коду набраної програми та її результати роботи (після усіх виправлень) представити у звіті

4. Виконати нижченаведені програми. Скріни коду набраних програм та її

їх результати роботи представити у звіті.

**Виконання роботи:**

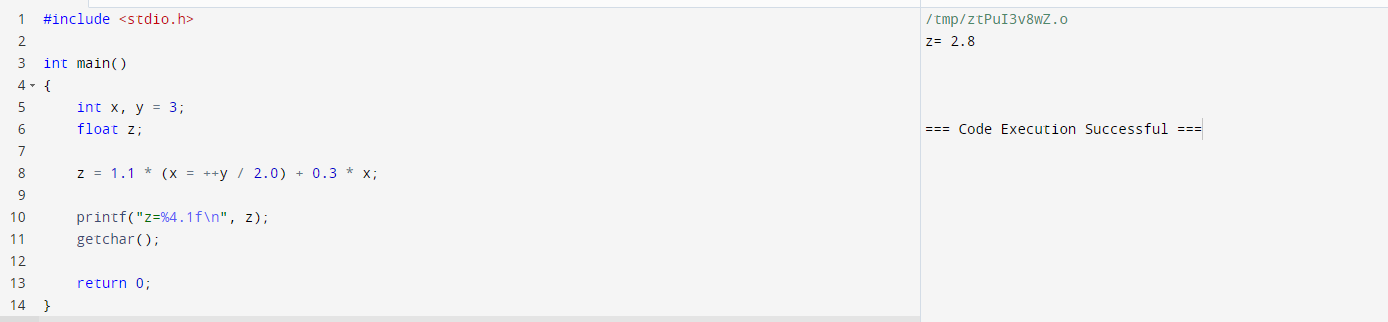
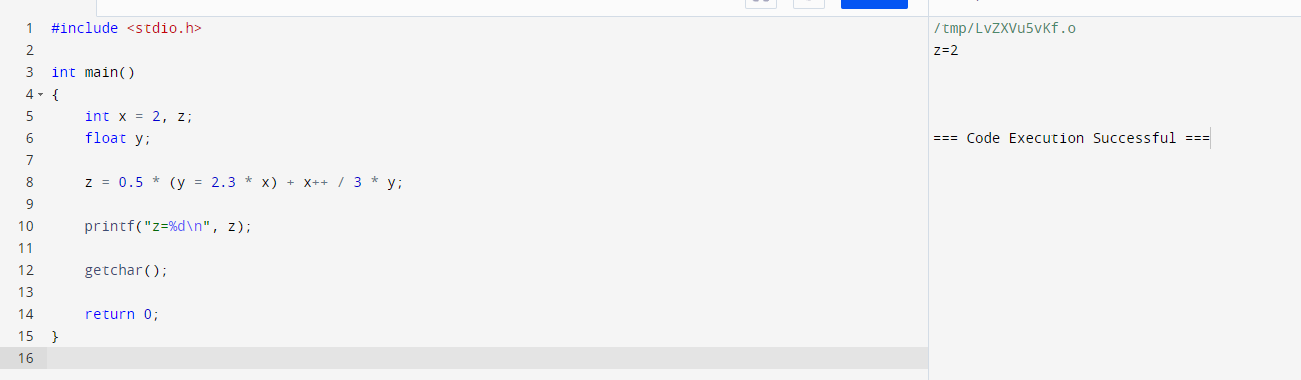
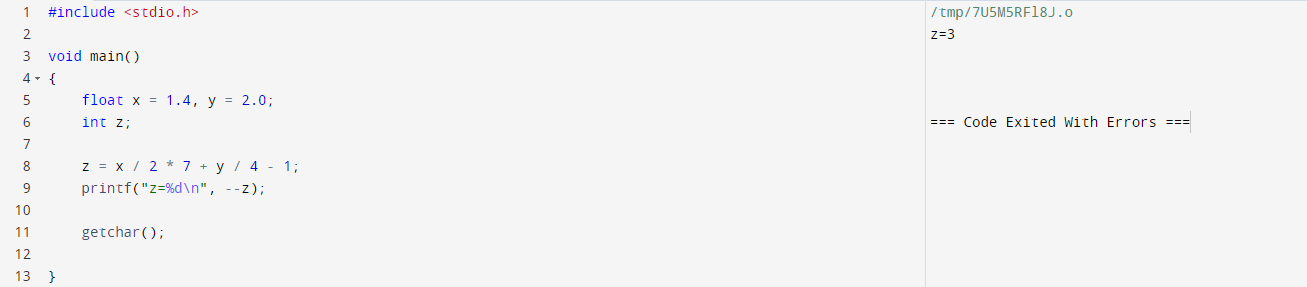
**Завдання 2**

****

**Завдання 3**

****

**Завдання 4**

****

**Контрольні запитання:**

1. Призначення та структура програми, написаної мовою C

Призначення:

Програми на мові C пишуться для виконання різних завдань, таких як обчислення, обробка даних, управління апаратним забезпеченням, розробка операційних систем, компіляторів, драйверів тощо. C є потужною та ефективною мовою програмування, яка дозволяє розробникам працювати на низькому рівні, що забезпечує високу продуктивність та ефективність.

Структура програми:

Типова програма на мові C складається з наступних компонентів:

- Препроцесорні директиви: Команди, які виконуються препроцесором перед компіляцією, наприклад, `#include`, `#define`.

- Функції: Основні блоки коду, які виконують певні дії. Кожна програма повинна містити функцію `main()`, з якої починається виконання програми.

- Оголошення змінних: Місце, де визначаються змінні, які будуть використовуватися у програмі.

- Оператори та вирази: Інструкції, які виконуються програмою для досягнення поставленої мети.

Приклад простої програми на C:

#include <stdio.h>

int main() {

printf("Hello, World!\n");

return 0;

}

2. Різновиди типів величин

В мові C існує кілька типів даних, які поділяються на:

- Примітивні типи:

- `int`: ціле число.

- `float`: число з плаваючою точкою.

- `double`: число з подвійною точністю.

- `char`: одиночний символ.

- `\_Bool`: логічний тип (булевий).

- Складні типи:

- Масиви: `int arr[10];`

- Структури: `struct { int a; float b; } myStruct;`

- Об'єднання: `union { int a; float b; } myUnion;`

- Перелічення: `enum { RED, GREEN, BLUE } color;`

- Указівники:

- `int \*ptr;` — указівник на ціле число.

3. Що таке константи і змінні?

- Константи: Це фіксовані значення, які не змінюються під час виконання програми. Константи можуть бути числовими, символьними або рядковими. В C їх можна визначати за допомогою ключового слова `const` або препроцесорної директиви `#define`.

#define PI 3.14159

const int MAX\_USERS = 100;

-Змінні: Це іменовані області пам'яті, які можуть змінювати своє значення під час виконання програми. Змінні мають тип, ім'я та значення.

int count = 10;

float price = 5.99;

```

4. Порядок виконання операцій

Порядок виконання операцій у C визначається пріоритетом та асоціативністю операторів:

- Пріоритет операторів(від найвищого до найнижчого):

1. `()`, `[]`, `.`, `->` (доступ до елементів)

2. `++`, `--` (постфіксні)

3. `+`, `-`, `\*`, `&`, `!`, `~` (унарні)

4. `\*`, `/`, `%` (множення, ділення, остача)

5. `+`, `-` (додавання, віднімання)

6. `<<`, `>>` (зсуви)

7. `<`, `<=`, `>`, `>=` (порівняння)

8. `==`, `!=` (рівність)

9. `&` (логічне І)

10. `^` (логічне виключне АБО)

11. `|` (логічне АБО)

12. `&&` (умовне І)

13. `||` (умовне АБО)

14. `? :` (тернарний оператор)

15. `=`, `+=`, `-=`, `\*=`, `/=`, `%=`, `&=`, `|=`, `^=`, `<<=`, `>>=` (присвоєння)

- Асоціативність:

- Зліва направо: більшість операторів (додавання, множення, порівняння тощо).

- Справа наліво: присвоєння, унарні оператори.

5. Особливості операцій інкремента і декремента

- Інкремент (`++`): Збільшує значення змінної на 1.

- Префіксна форма (`++i`): Збільшує значення змінної і повертає нове значення.

- Постфіксна форма (`i++`): Збільшує значення змінної, але повертає старе значення.

- Декремент (`--`): Зменшує значення змінної на 1.

- Префіксна форма (`--i`): Зменшує значення змінної і повертає нове значення.

- Постфіксна форма (`i--`): Зменшує значення змінної, але повертає старе значення.

6. Операції присвоєння

Операції присвоєння в мові C використовуються для збереження значень в змінних. Основний оператор присвоєння — це `=`, але існують також скорочені форми присвоєння, які комбінують інші операції з присвоєнням:

- Прості: `=`

int a = 5

- Комбіновані:

- `+=` (додавання з присвоєнням)

a += 2; // еквівалентно a = a + 2;

- `-=` (віднімання з присвоєнням)

a -= 3; // еквівалентно a = a - 3;

- `\*=` (множення з присвоєнням)

a \*= 4; // еквівалентно a = a \* 4

- `/=` (ділення з присвоєнням

a /= 2; // еквівалентно a = a / 2;

- `%=` остача з присвоєнням)

a %= 3; // еквівалентно a = a % 3;

7. Пояснити зміст і обгрунтувати результати виконаних прикладів

Приклад 1:

int a = 5;

int b = ++a; // a = 6, b = 6

int c = a++; // a = 7, c = 6

```

- Після `++a`, значення `a` збільшується на 1 і стає 6, а значення `b` теж стає 6.

- Після `a++`, значення `a` збільшується ще на 1 і стає 7, але значення `c` встановлюється в 6 (старе значення `a`).

Приклад 2:

int x = 10;

x += 5; // x = 15

x -= 3; // x = 12

x \*= 2; // x = 24

x /= 4; // x = 6

x %= 5; // x = 1

```

- Спочатку `x` дорівнює 10.

- Після `x += 5`, значення `x` стає 15.

- Після `x -= 3`, значення `x` стає 12.

- Після `x \*= 2`, значення `x` стає 24.

- Після `x /= 4`, значення `x` стає 6.

- Після `x %= 5`, значення `x` стає 1.