Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут телекомунікацій, радіоелектроніки та електронної техніки

кафедра «Радіоелектронні пристрої та системи»

Лабораторна робота №7

з дисципліни«Програмування частина 2»

«Арифметичні операції та вирази мови С»

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

Підготував:

ст. групи АП-11

Василюк Ростислав

Прийняв:

Чайковський І.Б.

Львів 2024р

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.

Проте, мова С має суттєвий недолік з точки зору потреб розробки радіотехнічних задач: тут недостатньо розвинені операції арифметики, зокрема, повністю відсутня комплексна арифметика, і її імітація призводить до генерування недостатньо ефективних кодів, що значно збільшує потреби часу при проведенні значних за обсягом математичних обчислень. Фірма Microsoft розробила власну версію мови С з інтерфейсом подібним до мови ФОРТРАН, найбільш пристосованою для математичних розрахунків і генеруючою найефективніші машинні коди. Паралельно на фірмі Borland велась розробка іншої версії мови С, перші варіанти якої мали назву “Turbo C", а пізніші - "Borland C", "C++", причому в версіях "C++" комплексну арифметику реалізують за допомогою класу об’єкта.

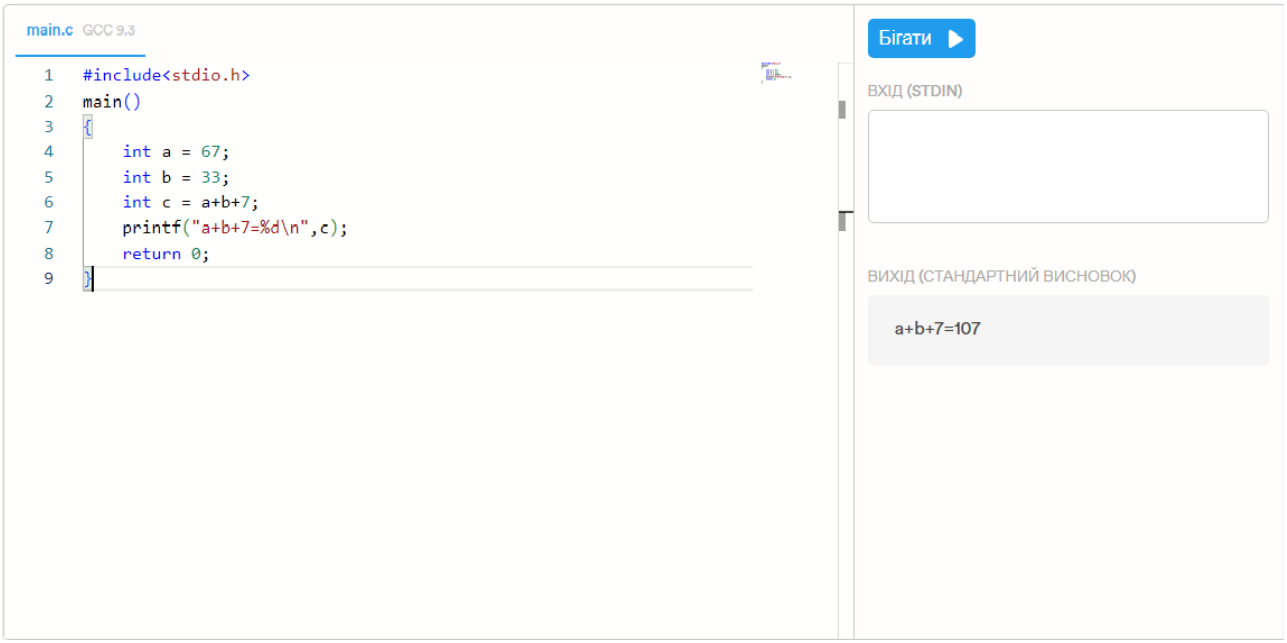
Файл - це просто поіменована область пам’яті на диску, в ньому може бути записана довільна інформація, в тому числі і програма.

Хід роботи

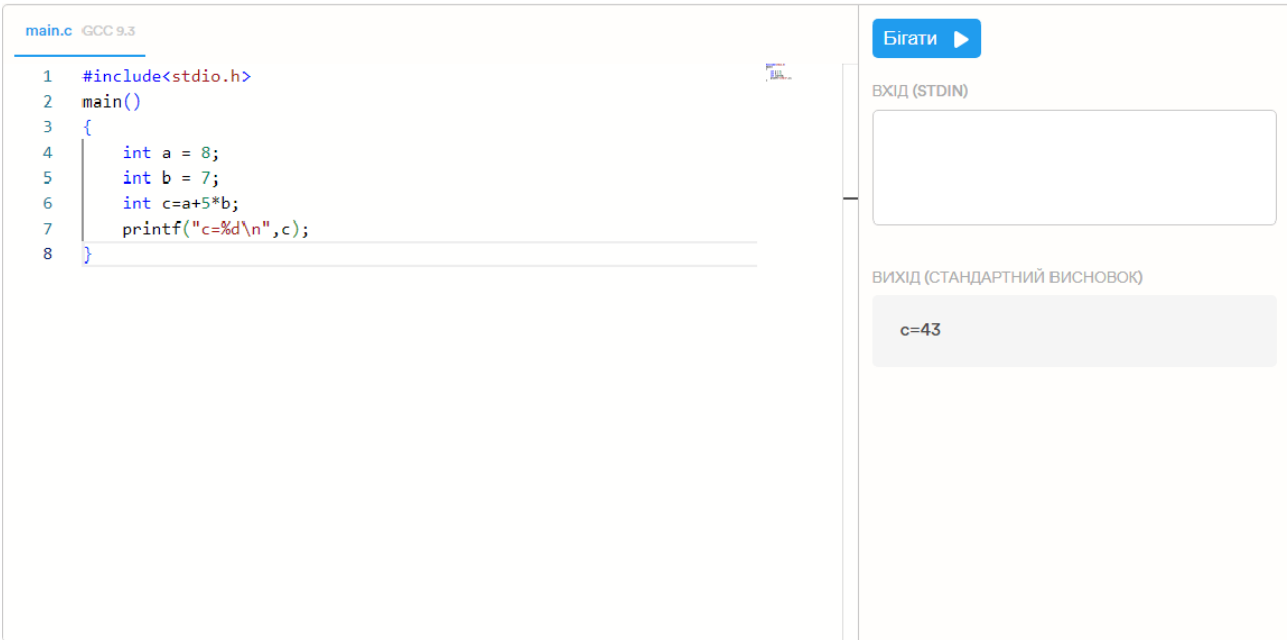
1. Ознайомитися з теоретичними відомостями.

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

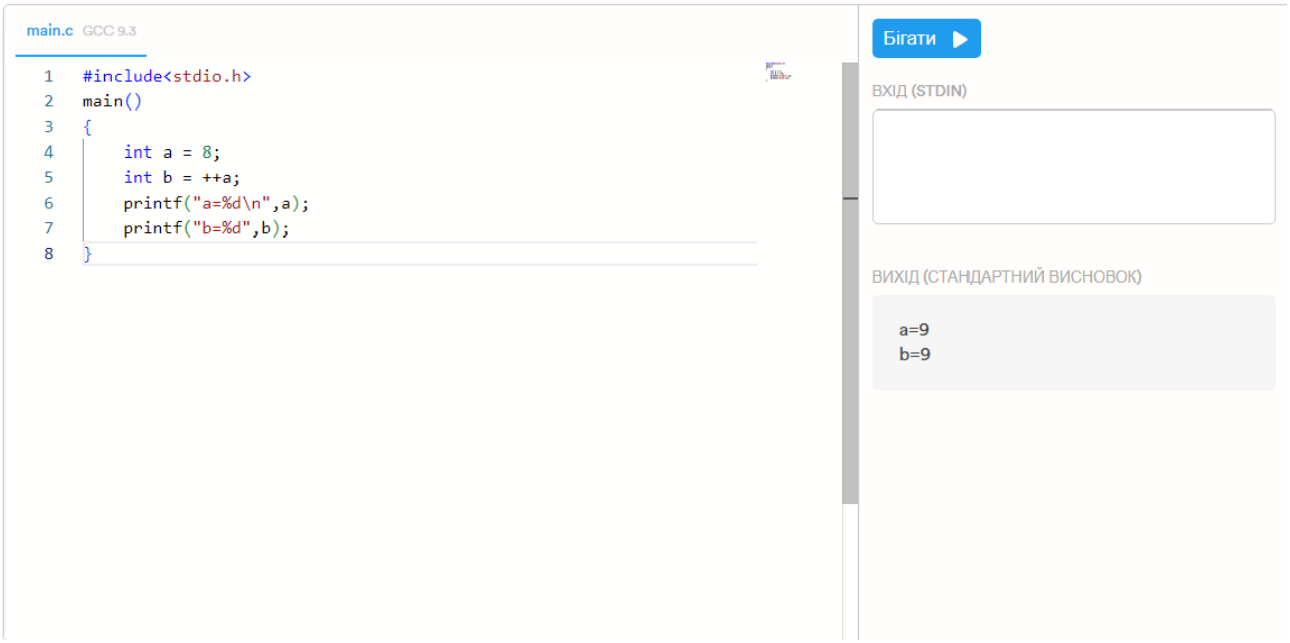
Розглянемо деякі приклади. Операція додавання повертає суму чисел.



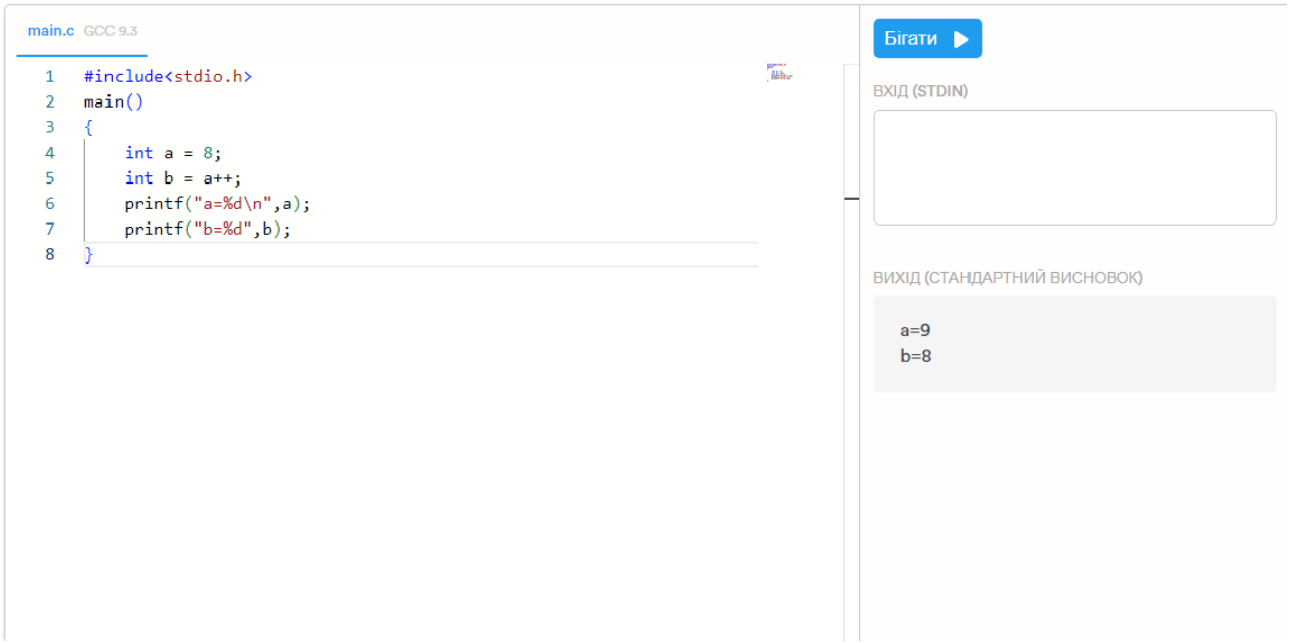
Операції *, /, % мають більший пріоритет, ніж + , - , і інші операції, відповідно до рис. 1.



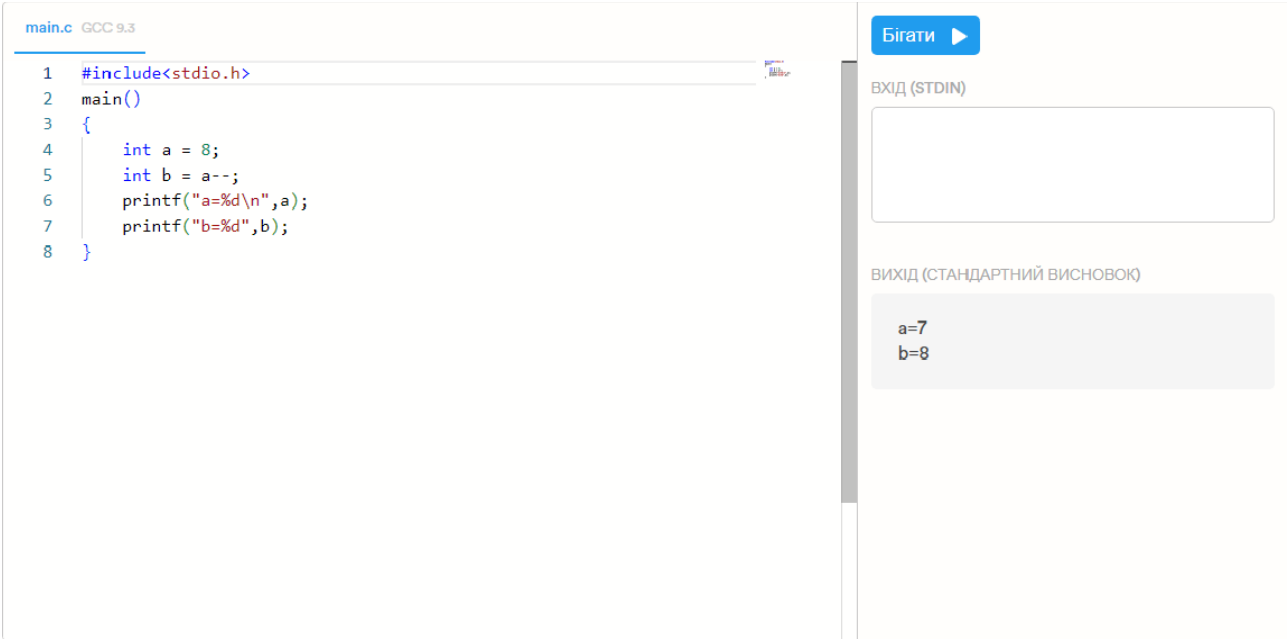
Також у С є спеціальні унарні операції над одним числом : ++ (інкремент) і – (декремент). Кожна з цих операцій має два види: префікс ний і постфіксний. Префіксний інкремент (++x) збільшує значення змінної на одиницю і отримане значення використовується як значення виразу ++x.



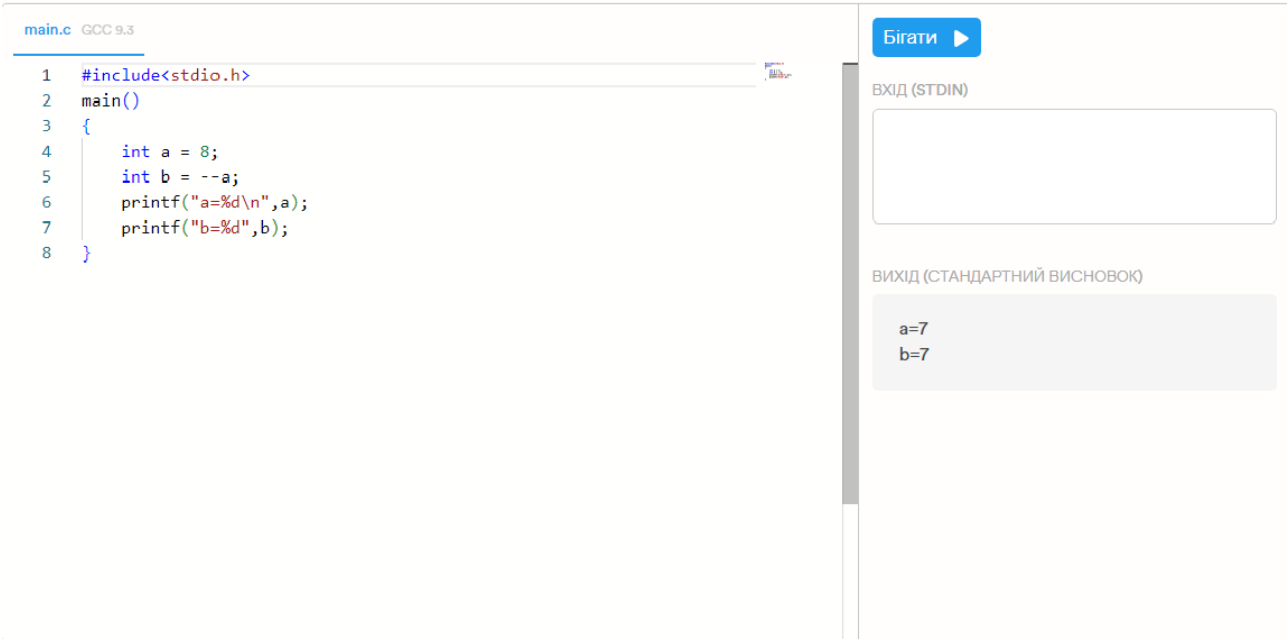
значенням виразу x++ буде те, яке було до збільшення на одиницю.



Постфіксний декремент (x--) зменшує значення змінної на одиницю, але значення виразу x-- буде те, яке було до зменшення на одиницю.



Префіксний декремент (--x) зменшує значення змінної на одиницю, і отримане значення використовується як значення виразу -- x.



3. Набрати текст нижченаведеної програми, виправити усі синтаксичні помилки, здійснити її компіляцію

```
#include <stdio.h> // Потрібно підключити заголовочний файл для використання функцій введення/виведення

#include <string.h> // Потрібно підключити заголовочний файл для використання функції strlen

#define PRAISE "О, яке чудове ім'я"

int main() // Потрібно визначити тип функції main та додати відкриваючу фігурну дужку

{

    char name[50];


    printf("Як Вас звати?\n");

    scanf("%s", name); // Виправлено помилку scnaf на scanf та додано пропущену лапку в специфікаторі формату


    printf("Привіт, %s. %s\n", name, PRAISE); // Виправлено друкарську помилку та додано пропущену кому


    printf("Ваше ім'я складається з %d літер і займає %d комірок пам'яті.\n", strlen(name), (int)sizeof(name)); // Додано відсутні аргументи та виправлено назву функції sizeof, додано приведення типу (int)


    printf("Вітальна фраза складається з %d літер і займає %d комірок пам'яті.\n", strlen(PRAISE), (int)sizeof(PRAISE)); // Додано відсутні аргументи та виправлено назву функції sizeof, додано приведення типу (int)


    return 0; // Повернення значення 0, оскільки функція main повертає ціле число

}
```

4.Виконати нижченаведені програми. Скріни коду набраних програм та її їх результати роботи представити у звіті.

1. #include<stdio.h>

#include<conio.h>

voild main()

{

float x=1.4,y=2.0;int z;

z=x/2*7+y/4-1;printf("z=%d\n",--z);

getch()

}

main.c:2:9: fatal error: conio.h: No such file or directory

2 | #include

| ^~~~~~~~

compilation terminated.

2. #include <stdio.h>

#include <conio.h>

void main()

{

int x = 2,2;

float y;

2=0.5*(y=2.3*x) +x++/3*y;

printf("z=%d\n",z);

```
getch();

}

#include <stdio.h>

#include <conio.h>

void main()

{

main.c:3:10: fatal error: conio.h: No such file or directory
    3 | #include
      |          ^~~~~~~~~
compilation terminated.
3. #include <stdio.h>

#include <conio.h>

void main()

{

int x,y=3;

float z:

z=1.1*(x+y/2.)+0.3*x;
```

```
printf("z=%4.1f\n",z),


getch();}
main.c:3:10: fatal error: conio.h: No such file or directory
   3 | #include
     |          ^~~~~~~~
compilation terminated.
```

Контрольні запитання:

1. Призначення та структура програми, написаної мовою С.

Мова програмування С використовується для розробки системного та застосункового програмного забезпечення. Основне призначення полягає в тому, щоб створювати ефективні та портативні програми.

Структура програми мовою С зазвичай складається з заголовочних файлів (якщо необхідно), об'яв функцій, функції main(), інструкцій, директив препроцесора та коментарів.

2. Різновиди типів величин.

Основні типи даних в мові С включають цілі числа (int), дійсні числа (float, double), символи (char), логічні значення (bool), вказівники (pointers), масиви (arrays) та структури (structures).

Крім того, в мові С є можливість створювати користувацькі типи даних за допомогою структур та об'єднань (unions).

3. Що таке константи і змінні?

Константами називаються елементи даних, яким присвоюються значення в описовій частині програми й у процесі виконання програми їх змінювати заборонено. Для визначення констант служить зарезервоване слово const. Змінні, на відміну від констант, можуть змінювати свої значення в процесі виконання програми.

4. Порядок виконання операцій.

В мові С порядок виконання операцій може бути визначений пріоритетами операторів. Наприклад, операції, які мають більший пріоритет, виконуються перед операціями з меншим пріоритетом. Також можна контролювати порядок виконання операцій за допомогою дужок.

5. Особливості операцій інкремента і декремента.

Оператор інкремента (++) збільшує значення змінної на одиницю.

Оператор декремента (--) зменшує значення змінної на одиницю.

Ці оператори можуть бути використані як префіксні (++i, --i), так і постфіксні (i++, i--) і можуть мати відмінність у виконанні в залежності від контексту використання.

При використанні як префіксні оператори збільшення або зменшення спочатку змінюють значення змінної, а потім повертають її нове значення.

При використанні як постфіксні оператори спочатку повертають поточне значення змінної, а потім змінюють його.

6. Операції присвоєння.

Присвоєння – механізм, що дозволяє змінювати значення об’єктів. Простими словами, за допомогою присвоєння можна задавати чи змінювати дані, які зберігаються у змінних. У мові С оператор присвоєння позначається символом дорівнює – “=”