

Міністерством освіти та науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт

про виконання лабораторної роботи №4
“ Основні типи змінних.Операції та їх пріоритет”

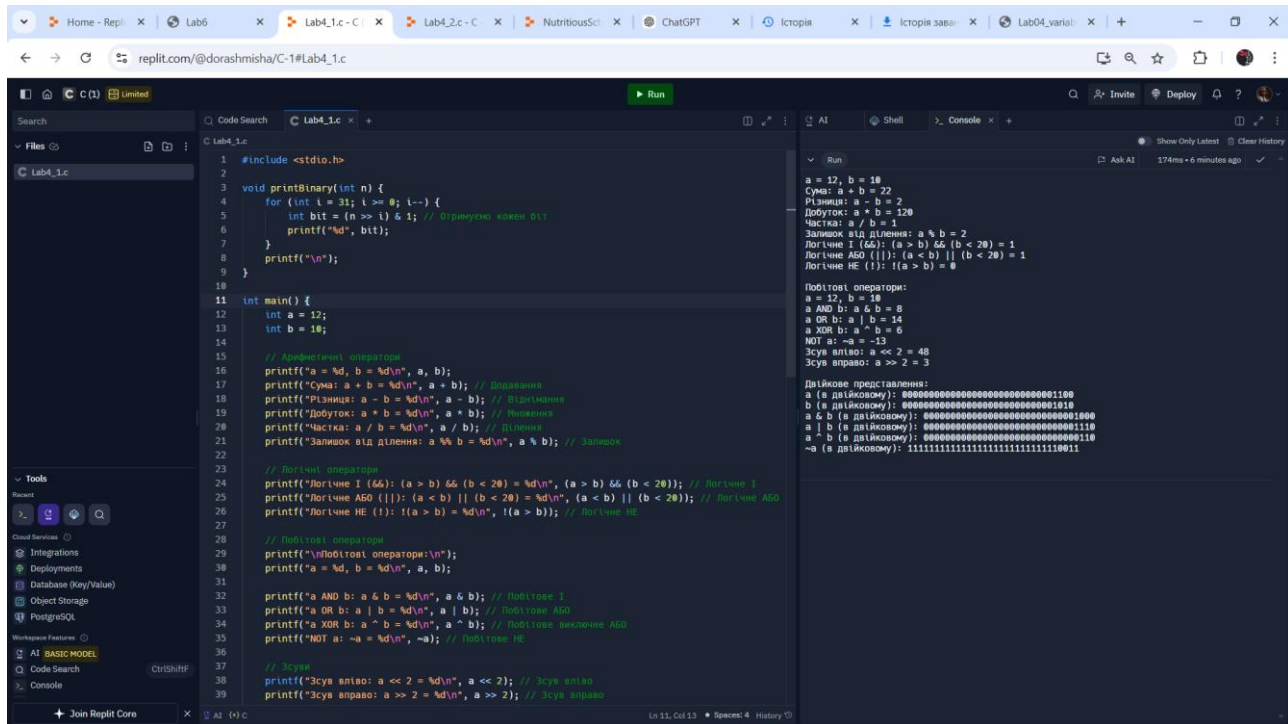
Виконав
студент групи ФЕП-11

Дораж Михайло

Перевірив:
Ас. Кужій Ю.

Мета роботи: Вивчити поняття: основні типи змінних, вказівники, операції та їх пріоритет.

1. Написати програму, котра ілюструє використання всіх арифметичних і логічних операторів мови Сі, для логічних побітових операцій використовувати представлення чисел в двійковому форматі.



```
1 #include <stdio.h>
2
3 void printBinary(int n) {
4     for (int i = 31; i >= 0; i--) {
5         int bit = (n >> i) & 1; // Візвачуємо кожен біт
6         printf("%d", bit);
7     }
8     printf("\n");
9 }
10
11 int main() {
12     int a = 12;
13     int b = 18;
14
15     // Арифметичні оператори
16     printf("a = %d, b = %d\n", a, b);
17     printf("Сума: a + b = %d\n", a + b); // Додавання
18     printf("Різниця: a - b = %d\n", a - b); // Віднімання
19     printf("Добуток: a * b = %d\n", a * b); // Множення
20     printf("Частка: a / b = %d\n", a / b); // Ділення
21     printf("Залишок від ділення: a %% b = %d\n", a % b); // Залишок
22
23     // Логічні оператори
24     printf("Логічне І (&&): (a > b) && (b < 20) = %d\n", (a > b) && (b < 20)); // Логічне І
25     printf("Логічне АБО (||): (a < b) || (b < 20) = %d\n", (a < b) || (b < 20)); // Логічне АБО
26     printf("Логічне НЕ (!): !(a > b) = %d\n", !(a > b)); // Логічне НЕ
27
28     // Побітові оператори
29     printf("\nПобітові оператори:\n");
30     printf("a = %d, b = %d\n", a, b);
31
32     printf("a AND b: a & b = %d\n", a & b); // Побітове І
33     printf("a OR b: a | b = %d\n", a | b); // Побітове АБО
34     printf("a XOR b: a ^ b = %d\n", a ^ b); // Побітове виключення АБО
35     printf("NOT a: ~a = %d\n", ~a); // Побітове НЕ
36
37     // Зсуви
38     printf("Зсув вліво: a << 2 = %d\n", a << 2); // Зсув вліво
39     printf("Зсув вправо: a >> 2 = %d\n", a >> 2); // Зсув вправо
40 }
```

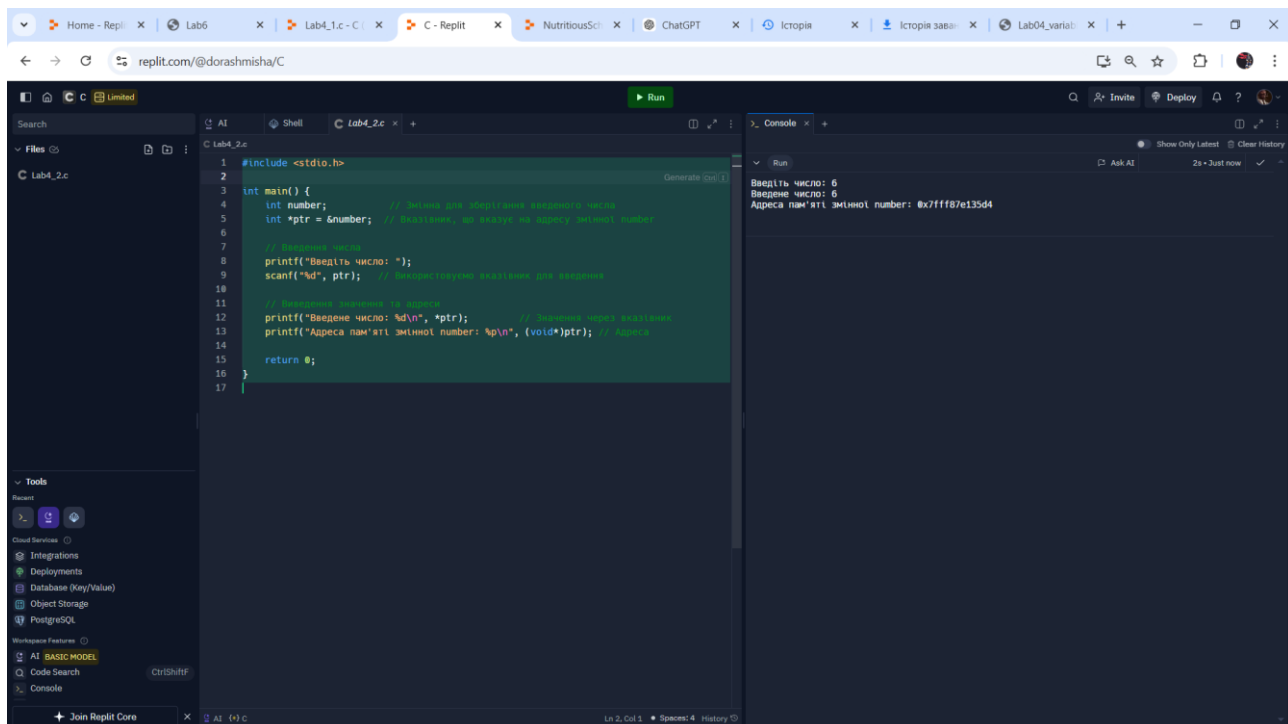
Run

a = 12, b = 18
Сума: a + b = 22
Різниця: a - b = 2
Добуток: a * b = 120
Частка: a / b = 1
Залишок від ділення: a % b = 2
Логічне І (&&): (a > b) && (b < 20) = 1
Логічне АБО (||): (a < b) || (b < 20) = 1
Логічне НЕ (!): !(a > b) = 0

Побітові оператори:
a = 12, b = 18
a AND b: a & b = 8
a OR b: a | b = 14
a XOR b: a ^ b = 6
NOT a: ~a = -13
Зсув вліво: a << 2 = 48
Зсув вправо: a >> 2 = 3

Двійкове представлення:
a (в двійковому): 00000000000000000000000000001100
b (в двійковому): 00000000000000000000000000010010
a & b (в двійковому): 00000000000000000000000000010000
a | b (в двійковому): 00000000000000000000000000011110
a ^ b (в двійковому): 00000000000000000000000000011000
~a (в двійковому): 11111111111111111111111111110011

2. Написати програму з введенням числа у змінну і виведенням її адреси та значення через вказівник



```
1 #include <stdio.h>
2
3 int main() {
4     int number; // Змінна для зберігання введенного числа
5     int *ptr = &number; // Вказівник, що вказує на адресу змінної number
6
7     // Введення числа
8     printf("Введіть число: ");
9     scanf("%d", ptr); // Використання вказівника для введення
10
11     // Виведення значення та адреси
12     printf("Введене число: %d\n", *ptr); // Виведення через вказівник
13     printf("Адреса пам'яті змінної number: %p\n", (void*)ptr); // Адреса
14
15     return 0;
16 }
17
```

Run

Введіть число: 6
Введене число: 6
Адреса пам'яті змінної number: 0x7fff07e135d4

3. Написати програму розв'язку квадратного рівняння, коефіцієнти якого вводяться з клавіатури.

```
25
26 // Перевірка значення дискримінанту
27 if (discriminant > 0) {
28     // Два різних дійсних корені
29     root1 = (-b + sqrt(discriminant)) / (2 * a);
30     root2 = (-b - sqrt(discriminant)) / (2 * a);
31     printf("Два різних дійсних корені: x1 = %.2lf, x2 = %.2lf\n", root1, root2);
32 } else if (discriminant == 0) {
33     // Один дійсний корінь
34     root1 = -b / (2 * a);
35     printf("Один дійсний корінь: x = %.2lf\n", root1);
36 } else {
37     // Комплексні корені
38     double realPart = -b / (2 * a);
39     double imaginaryPart = sqrt(-discriminant) / (2 * a);
40     printf("Комплексні корені: x1 = %.2lf + %.2lfi, x2 = %.2lf - %.2lfi\n",
41           realPart, imaginaryPart, realPart, imaginaryPart);
42 }
43 return 0;
44 }
45
```

Console output:

```
Введіть коефіцієнт a (не може бути 0): 2
Введіть коефіцієнт b: -3
Введіть коефіцієнт c: 1
Два різних дійсних корені: x1 = 1.00, x2 = 0.50
```

Висновки: Лабораторна робота № 4 допомогла освоїти основні типи змінних, оператори та вказівники у мові Сі. Реалізація програми для розв'язання квадратного рівняння підтвердила важливість введення даних і математичних обчислень, що сприяло розвитку практичних навичок у програмуванні