Звіт

# Виконання лабораторної роботи №2

Студента 2ПІ-25Б Заболотного Олександра Івановича

«Цикли»

Тема: Цикли

Мета: навчитися створювати з використанням   
циклів (for, while, do-while)

Завдання 8. Вивести всі дільники введеного цілого числа.

Текст програми:

#include <stdio.h>

int main() {

    int n;

    printf("Enter an integer: ");

    scanf("%d", &n);

    printf("Divisors of %d: ", n);

    for (int i = 1; i <= n; i++) {

        if (n % i == 0) {

            printf("%d ", i);

        }

    }

    printf("\n");

    return 0;

}

1. Призначення програми:

Програма призначена для знаходження всіх дільників введеного користувачем цілого числа. Вона перебирає всі можливі дільники та виводить ті, на які число ділиться без остачі.

2. Вхідні дані:

Ціле число n (вводиться користувачем).

3. Алгоритм роботи програми:

• Введення цілого числа n.

• Перебір всіх чисел від 1 до n.

• Для кожного числа перевірка, чи є воно дільником (n % i == 0).

• Виведення всіх знайдених дільників.

4. Опис основних частин коду:

• #include <stdio.h> — підключення бібліотеки для введення/виведення.

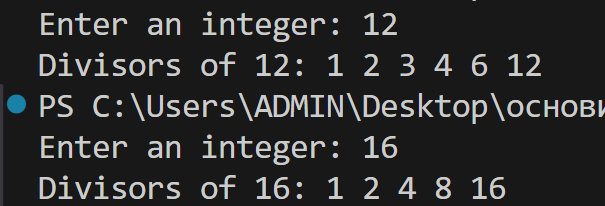
• Оголошення змінної для збереження числа n.

• Введення значення n через scanf.

• Цикл for для перебору всіх можливих дільників.

• Умовний оператор if для перевірки дільника.

• Виведення результату на екран.

Результат роботи:  


Висновок: У ході виконання лабораторної роботи я ознайомився з основами програмування на мові C, навчився працювати з компілятором та середовищем розробки. Я навчився оголошувати змінні, використовувати оператори введення та виведення, а також цикли і умовні конструкції для реалізації алгоритмів.

В результаті я створив програму, яка знаходить усі дільники введеного цілого числа.

Отримані знання та навички стануть основою для подальшого вивчення програмування та розробки більш складних програм.

Контрольні питання:  
8. Правила перетворення типів у мові C:

Автоматичне перетворення (неявне):

При арифметичних операціях тип з меншою розрядністю автоматично перетворюється до типу з більшою розрядністю

char → int → long → float → double → long double

При присвоєнні значення автоматично перетворюється до типу змінної

Явне перетворення (приведення типу):

Використовується оператор (тип)значення, наприклад: (int)3.14

9. Пріоритет виконання операцій у мові C (від найвищого до найнижчого):

1. Дужки (), індексація [], доступ до члена структури ., ->
2. Унарні оператори !, ~, ++, --, +, -, \* (розіменування), & (адреса)
3. Мультиплікативні \*, /, %
4. Адитивні +, -
5. Зсув <<, >>
6. Порівняння <, <=, >, >=
7. Рівність ==, !=
8. Побітове "і" &
9. Побітове "виключне або" ^
10. Побітове "або" |
11. Логічне "і" &&
12. Логічне "або" ||
13. Умовний оператор ?:
14. Присвоєння =, +=, -=, тощо
15. Кома ,

10. Робота оператора розгалуження if:

Синтаксис: if (умова) { оператори; } else { оператори; }

Принцип роботи:

Перевіряється умова у дужках

Якщо умова істинна (не дорівнює 0), виконується блок після if

Якщо умова хибна (дорівнює 0), виконується блок після else (якщо він є)

Можна використовувати else if для перевірки декількох умов послідовно