

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра КІТАМ
Дисципліна «Програмування»

Звіт

з лабораторної роботи №2
«Директиви препроцесора і функції printf() і scanf()»

Виконала:
ст.гр. АКТСІз-23-1
Рацебарська А.Д.

Прийняла:
доц. Максимова С.С.,

Харків 2023

2. ДИРЕКТИВИ ПРЕПРОЦЕСОРА І ФУНКЦІЇ PRINTF() І SCANF()

2.1 Мета роботи:

Вивчити особливості роботи директив препроцесора і функцій printf() і scanf().

2.2 Теоретичні відомості:

Директиви препроцесора у C++

- Директиви препроцесора в мові програмування C++ використовуються для вказівок препроцесору перед початком компіляції коду.
 - *Препроцесор* - це програма, яка обробляє вихідний код до його компіляції.
- Одними з найбільш часто використовуваних директив препроцесора є *#define*, *#if*, *#else*, *#elif* та *#endif*.

#define використовується для визначення символічних констант.

#if, *#else*, *#elif* та *#endif* використовуються для умовної компіляції. Це дозволяє включати або виключати частини коду в залежності від певних умов.

Функції printf() і scanf()

printf() та *scanf()* є стандартними функціями вводу/виводу в C++, успадкованими від мови C. Вони використовуються для виведення на екран та введення даних від користувача відповідно.

- *printf()* використовується для форматowanego виводу. Ця функція дозволяє виводити текст та змінні різних типів, використовуючи **специфікатори формату**, такі як *%d* для цілих чисел, *%f* для чисел з плаваючою комою тощо.
- *scanf()* використовується для форматowanego вводу. Ця функція дозволяє зчитувати дані різних типів, вказаних за допомогою специфікаторів формату, і зберігати їх у відповідних змінних.

Директиви препроцесора у C++ (доповнення)

Директиви препроцесора в C++ мають широке використання. Окрім умовної компіляції, вони включають директиви для включення файлів, наприклад, *#include*, що дозволяє включити зміст іншого файлу під час компіляції. Це особливо корисно для організації коду та уникнення його дублювання.

Інша важлива директива - це *#ifdef*, яка дозволяє перевірити, чи була визначена певна макрокоманда, та *#ifndef*, що робить протилежне, - перевіряє, чи не була визначена макрокоманда. Це часто використовується для уникнення проблем з подвійним включенням заголовочних файлів.

Функції printf() і scanf() у C++ (доповнення)

printf() та *scanf()* не лише дозволяють здійснювати базовий ввід-вивід, але й забезпечують значну гнучкість у форматуванні даних. Наприклад, *printf()* може використовувати широкий спектр специфікаторів формату для виводу чисел з різною

точністю, виводу тексту у різних форматах (наприклад, шістнадцятковому) та навіть для вирівнювання тексту та чисел.

З іншого боку, `scanf()` забезпечує можливість зчитування різних типів даних, включаючи цілі числа, числа з плаваючою комою, символи та рядки. Вона також дозволяє визначати формат введення, що робить її потужним інструментом для обробки вводу користувача.

2.3 Хід роботи:

Варіант 2

Виконати:

1. Програмування директив препроцесора: За допомогою директив `#if`, `#else`, `#elif` здійснити вибір рядків програми для обчислення $2(a + b)$, або $a * b$.
2. Програмування функцій `printf()` і `scanf()`: Ввести два цілочисельних значення і вивести їх приватно на екран монітора.

```
12
13 #include <iostream>
14 #include <cstdio> // Для використання printf і scanf
15
16 #define CALC_MODE 1 // Змініть на 0 або 1 для вибору різних частин коду
17
18 int main() {
19     int a, b;
20
21     // Введення значень a і b
22     printf("Введіть два цілі числа:\n");
23     scanf("%d %d", &a, &b);
24
25     // Використання директив препроцесора для вибору операції
26     #if CALC_MODE == 1
27         int result = 2 * (a + b);
28         printf("Результат виразу 2 * (a + b): %d\n", result);
29     #else
30         int result = a * b;
31         printf("Результат виразу a * b: %d\n", result);
32     #endif
33
34     return 0;
35 }
36
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● alonaratsebarska@MacBook-Pro-Alona first_academic_year % g++ -o 2_laboratory 2_laboratory.cpp
● alonaratsebarska@MacBook-Pro-Alona first_academic_year % ./2_laboratory
Введіть два цілі числа:
5
7
Результат виразу a * b: 35
● alonaratsebarska@MacBook-Pro-Alona first_academic_year % g++ -o 2_laboratory 2_laboratory.cpp
● alonaratsebarska@MacBook-Pro-Alona first_academic_year % ./2_laboratory
Введіть два цілі числа:
5
7
Результат виразу 2 * (a + b): 24
○ alonaratsebarska@MacBook-Pro-Alona first_academic_year %
```

У цьому кодї:

1. `#define CALC_MODE 1` визначає константу препроцесора. Якщо її значення 1, програма виконує обчислення $2 * (a + b)$. Якщо її змінити на 0, програма обчислюватиме $a * b$.
2. Функції `printf()` і `scanf()` використовуються для вводу та виводу цілих чисел.

ВИСНОВКИ

У процесі виконання цієї лабораторної роботи були засвоєні ключові аспекти використання директив препроцесора та функцій `printf()` і `scanf()`. Навчилася ефективно використовувати директиви препроцесора для умовної компіляції коду, що дозволяє змінювати поведінку програми без зміни самого коду. Також оволоділа навичками форматованого вводу та виводу даних, що є важливим для створення взаємодії з користувачем у консольних програмах. Написала програму, що демонструє застосування цих елементів для розрахунку виразів залежно від заданих умов.