# Practical-1

**Aim: Setup VS Code + GCC, write the first C program, explain the structure, Print name & age using printf**

**Steps to install and set up the Compiler are:**

1. Go to any browser and download the exe file of Mingw.
2. The path for installation should be C:\mingw
3. Click install
4. Make sure to mark gcc compiler and the basic MinGW setup for installation
5. Click on Apply Changes

**Add compiler path**

1. Open the edit system environment variables
2. Select the path, click edit
3. Click on new and enter this path C:\mingw\bin
4. Select ok to apply the changes
5. Run gcc --version to verify the compiler path is set correctly.

**Install VS Code**

1. Go to any browser and download the exe file of VS Code
2. Click I accept the agreement
3. Put VS Code for installation by Microsoft

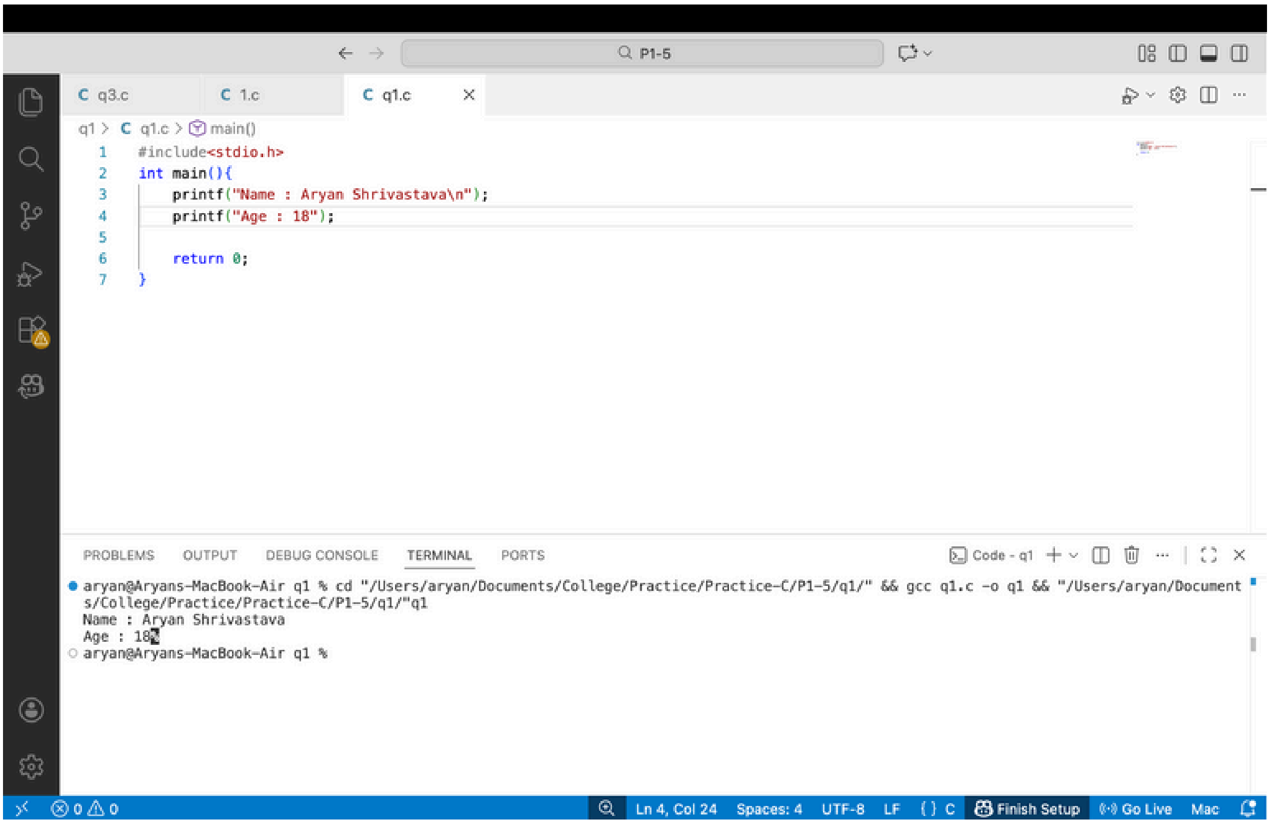
**Set up VS Code**

1. Dowload C / C++ extension pack by Microsoft.
2. Download Code Runner extension.
3. Open settings, select run in terminal.

**Steps to write the first code**

1. Create a folder and open it in VS Code
2. Create a new file, myinfo.c
3. Write code

**Code and output:**



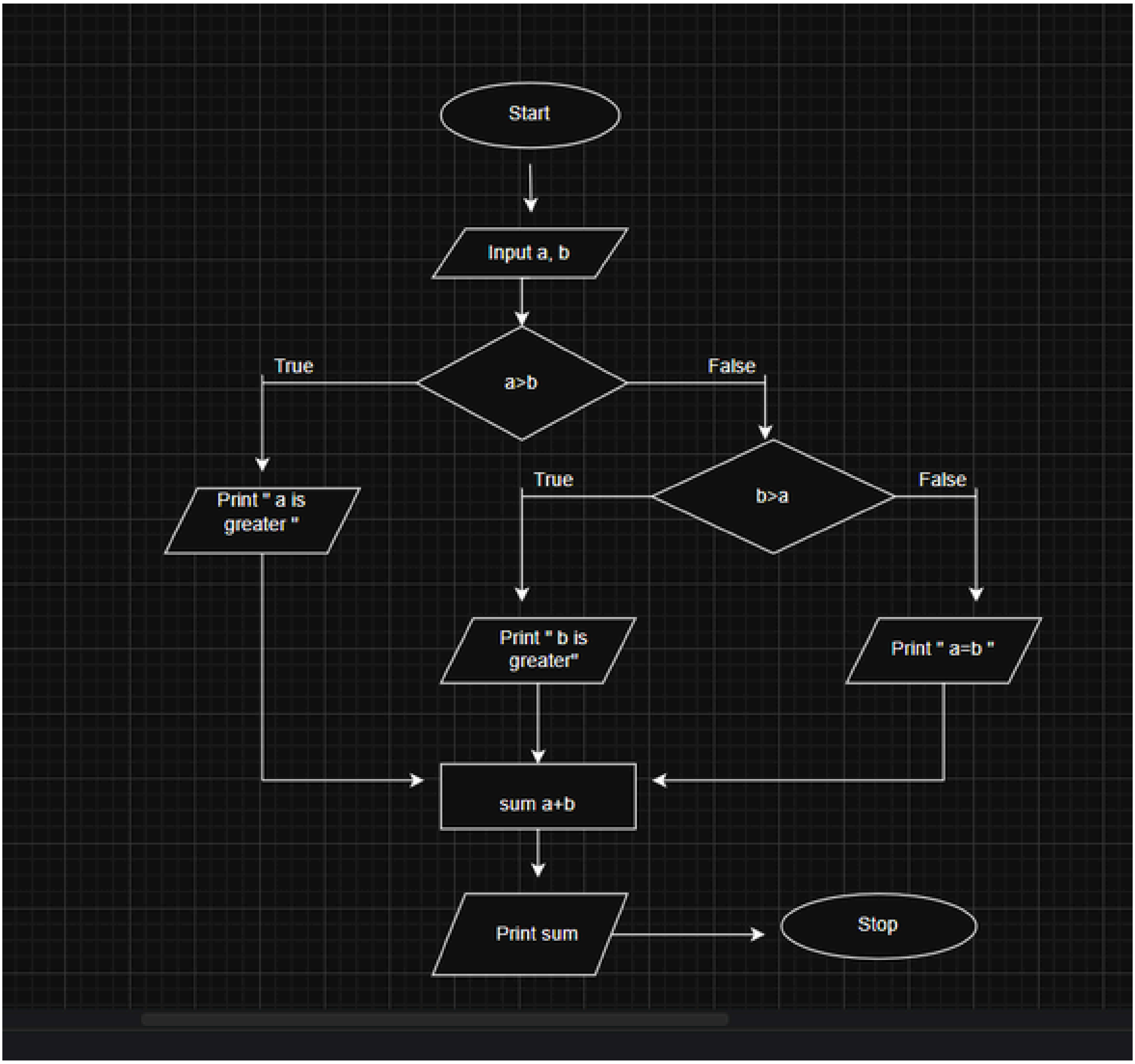
```c
#include<stdio.h>
int main(){
    printf("Name : Aryan Shrivastava\n");
    printf("Age : 18");

    return 0;
}
```

```
aryan@Aryans-MacBook-Air q1 % cd "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q1/" && gcc q1.c -o q1 && "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q1/"q1
Name : Aryan Shrivastava
Age : 18%
aryan@Aryans-MacBook-Air q1 %
```

# Practical-2

**Aim: Draw flowcharts for sum & largest of two numbers, implement in C, and show VS Code shortcuts.**

**Flowchart for showing the largest of two numbers and their sum:**



## Algorithm:

**Step 1:** Start

**Step 2:** Input two numbers a and b

**Step 3:** Compute sum = a + b

**Step 4:** Check if a == b

If true, then

➤ Display “Both numbers are equal”

Else

➤ If a > b then

→ largest = a

→ Display “Largest = a”

➤ Else

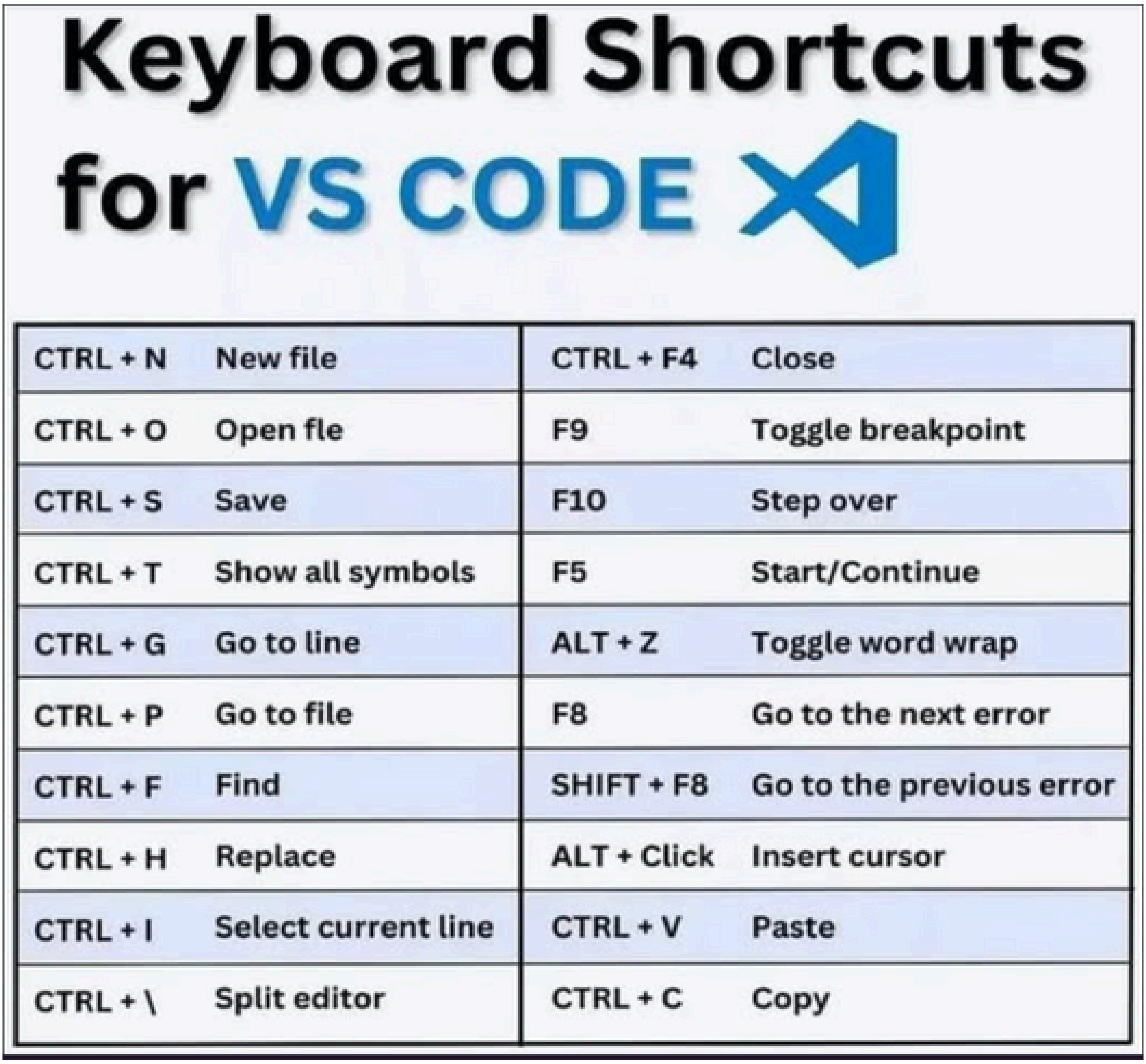
→ largest = b

→ Display “Largest = b”

**Step 5:** Display sum

**Step 6:** Stop

# VS code shortcuts:



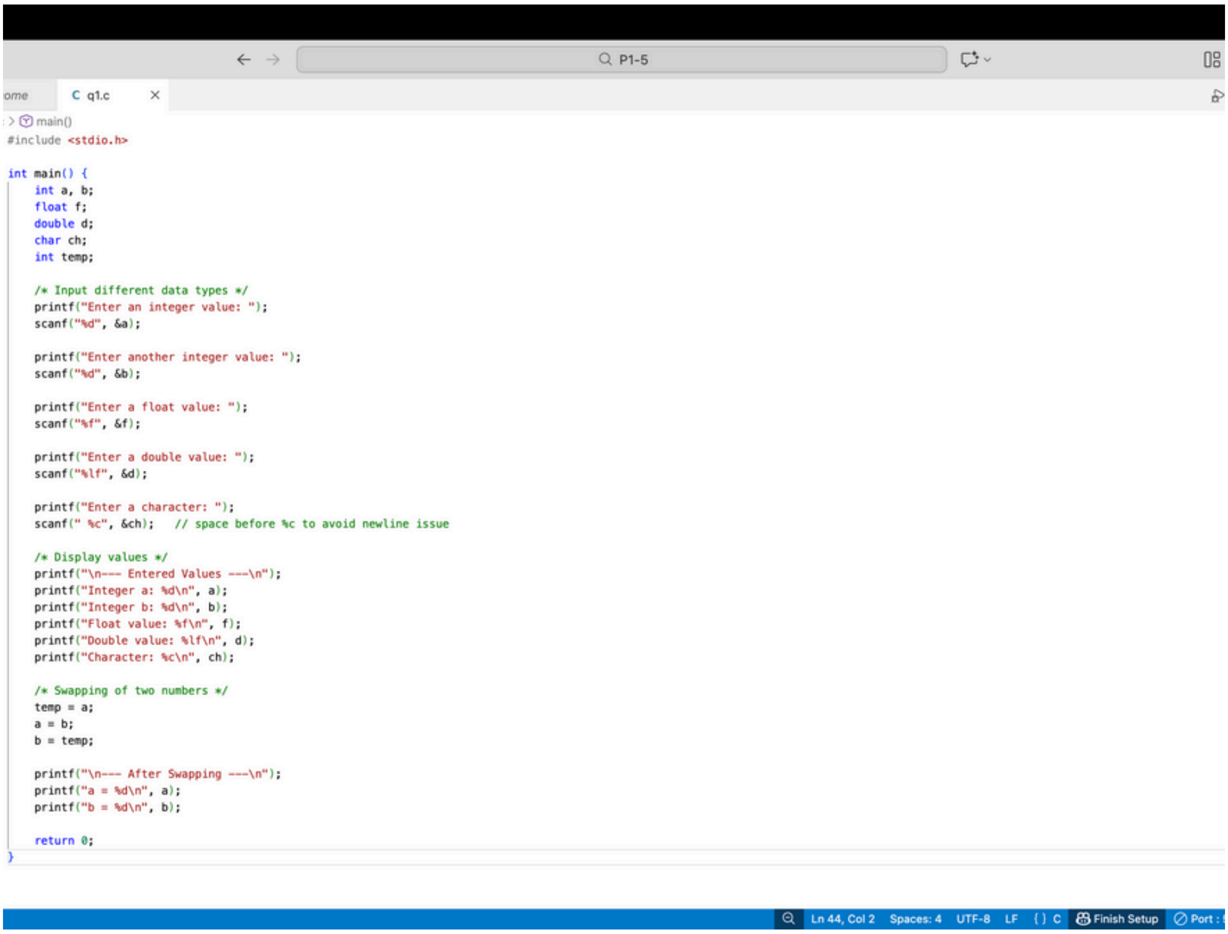
# Keyboard Shortcuts for VS CODE

| | | | |
|---|---|---|---|
| CTRL + N | New file | CTRL + F4 | Close |
| CTRL + O | Open fle | F9 | Toggle breakpoint |
| CTRL + S | Save | F10 | Step over |
| CTRL + T | Show all symbols | F5 | Start/Continue |
| CTRL + G | Go to line | ALT + Z | Toggle word wrap |
| CTRL + P | Go to file | F8 | Go to the next error |
| CTRL + F | Find | SHIFT + F8 | Go to the previous error |
| CTRL + H | Replace | ALT + Click | Insert cursor |
| CTRL + I | Select current line | CTRL + V | Paste |
| CTRL + \ | Split editor | CTRL + C | Copy |

## Practical-3

**Aim: Use of various data types along with format specifiers and swapping of two numbers.**

**Code:**



```c
#include <stdio.h>

int main() {
    int a, b;
    float f;
    double d;
    char ch;
    int temp;

    /* Input different data types */
    printf("Enter an integer value: ");
    scanf("%d", &a);

    printf("Enter another integer value: ");
    scanf("%d", &b);

    printf("Enter a float value: ");
    scanf("%f", &f);

    printf("Enter a double value: ");
    scanf("%lf", &d);

    printf("Enter a character: ");
    scanf(" %c", &ch);   // space before %c to avoid newline issue

    /* Display values */
    printf("\n--- Entered Values ---\n");
    printf("Integer a: %d\n", a);
    printf("Integer b: %d\n", b);
    printf("Float value: %f\n", f);
    printf("Double value: %lf\n", d);
    printf("Character: %c\n", ch);

    /* Swapping of two numbers */
    temp = a;
    a = b;
    b = temp;

    printf("\n--- After Swapping ---\n");
    printf("a = %d\n", a);
    printf("b = %d\n", b);

    return 0;
}
```

# Practical-4

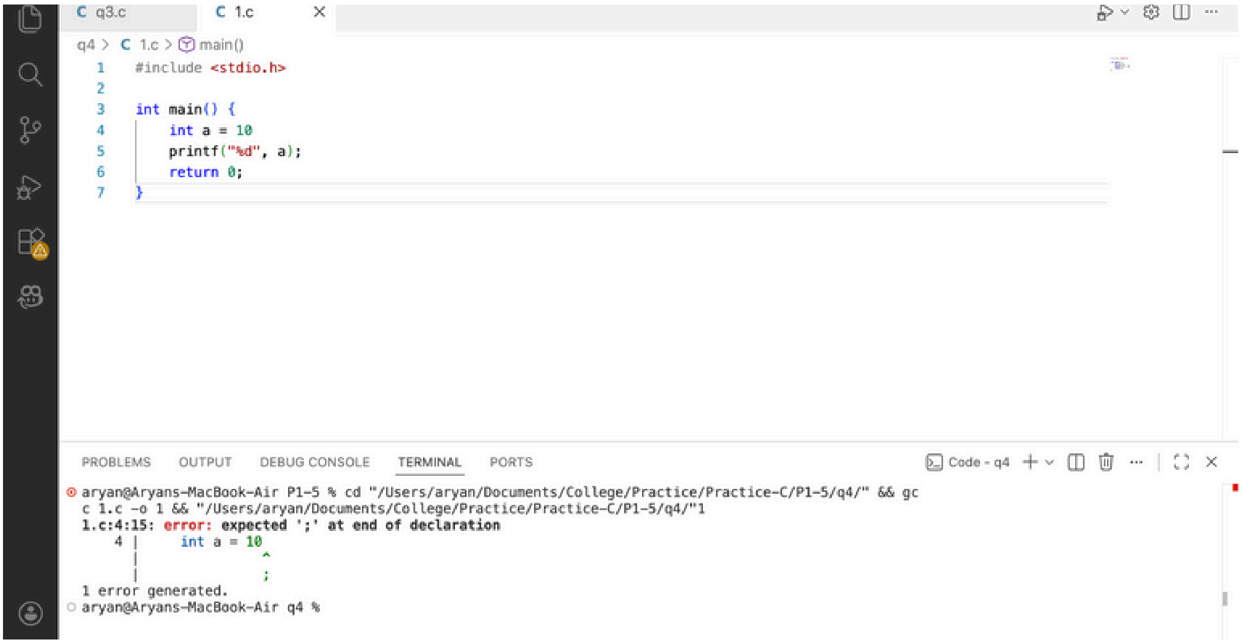
## Aim: Explanation of compile-time and run-time errors.

1. **Compile-time errors:** A compile-time error happens when you write incorrect code that the compiler cannot translate into an executable program. These errors are detected before the program runs, during the compilation stage.

## Types of compile-time errors:

- **Syntax Errors:** A syntax error occurs when a statement in a C program violates the grammatical or structural rules of the language, preventing successful compilation.

**Example:**

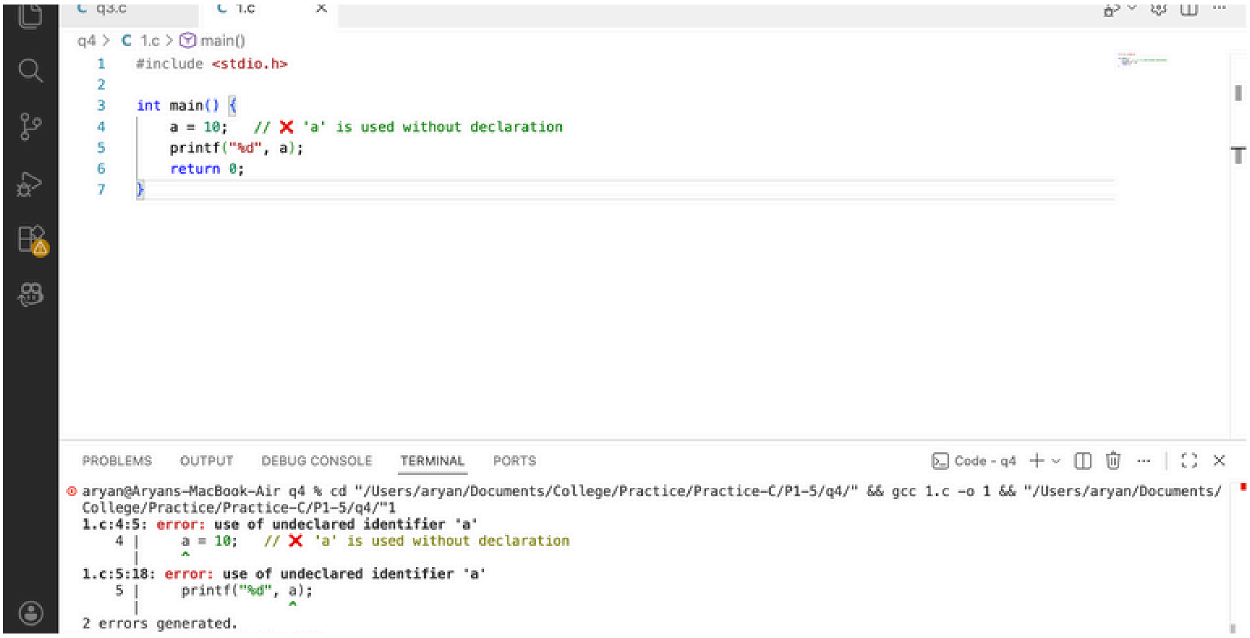


```c
#include <stdio.h>

int main() {
    int a = 10
    printf("%d", a);
    return 0;
}
```

```
aryan@Aryans-MacBook-Air P1-5 % cd "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q4/" && gc
c 1.c -o 1 && "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q4/"1
1.c:4:15: error: expected ';' at end of declaration
    4 |     int a = 10
      |               ^
      |               ;
1 error generated.
aryan@Aryans-MacBook-Air q4 %
```

- **Undeclared Variable Errors:** These occur when a variable is used in a program before it has been declared or defined.

**Example:**

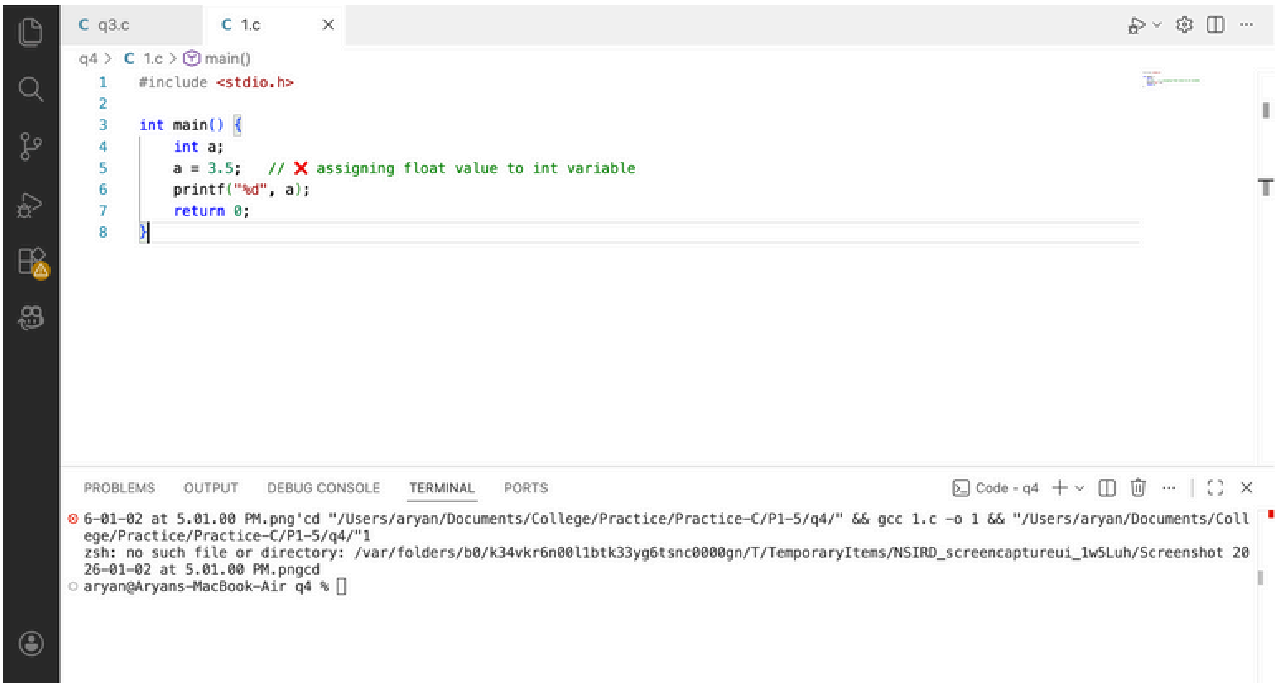


```c
#include <stdio.h>

int main() {
    a = 10;   // ❌ 'a' is used without declaration
    printf("%d", a);
    return 0;
}
```

```
aryan@Aryans-MacBook-Air q4 % cd "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q4/" && gcc 1.c -o 1 && "/Users/aryan/Documents/
College/Practice/Practice-C/P1-5/q4/"1
1.c:4:5: error: use of undeclared identifier 'a'
    4 |     a = 10;   // ❌ 'a' is used without declaration
      |     ^
1.c:5:18: error: use of undeclared identifier 'a'
    5 |     printf("%d", a);
      |                  ^
2 errors generated.
```

- **Type Mismatch Errors:** These occur when an operation or assignment involves incompatible data types, such as assigning a value of one type to a variable of another type.

**Example:**

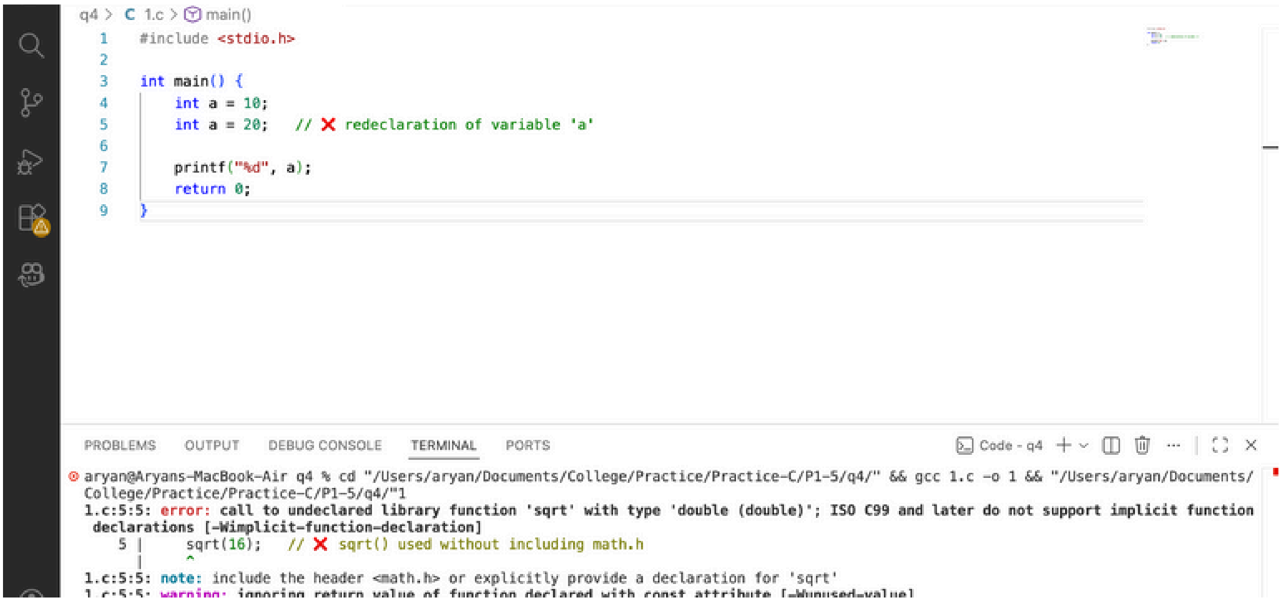


```c
#include <stdio.h>

int main() {
    int a;
    a = 3.5;    // ❌ assigning float value to int variable
    printf("%d", a);
    return 0;
}
```

```
⊗ 6-01-02 at 5.01.00 PM.png'cd "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q4/" && gcc 1.c -o 1 && "/Users/aryan/Documents/Coll
  ege/Practice/Practice-C/P1-5/q4/"1
  zsh: no such file or directory: /var/folders/b0/k34vkr6n00l1btk33yg6tsnc0000gn/T/TemporaryItems/NSIRD_screencaptureui_1w5Luh/Screenshot 20
  26-01-02 at 5.01.00 PM.pngcd
○ aryan@Aryans-MacBook-Air q4 % []
```

- **Redeclaration Error:** This compile-time error occurs when a variable or function is declared more than once with the same name in the same scope, causing a conflict for the compiler.

**Example:**



```c
#include <stdio.h>

int main() {
    int a = 10;
    int a = 20;    // ❌ redeclaration of variable 'a'

    printf("%d", a);
    return 0;
}
```

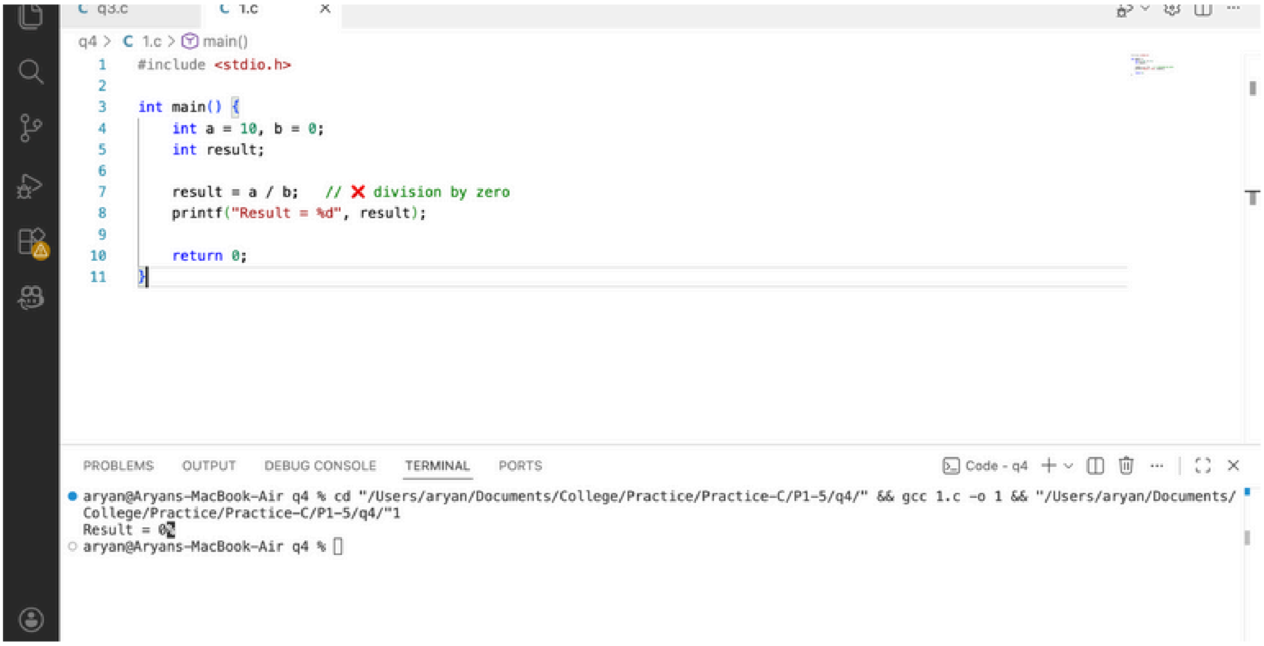
```
aryan@Aryans-MacBook-Air q4 % cd "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q4/" && gcc 1.c -o 1 && "/Users/aryan/Documents/
College/Practice/Practice-C/P1-5/q4/"1
1.c:5:5: error: call to undeclared library function 'sqrt' with type 'double (double)'; ISO C99 and later do not support implicit function
 declarations [-Wimplicit-function-declaration]
    5 |     sqrt(16);    // ❌ sqrt() used without including math.h
      |     ^
1.c:5:5: note: include the header <math.h> or explicitly provide a declaration for 'sqrt'
1.c:5:5: warning: ignoring return value of function declared with const attribute [-Wunused-value]
```

1. **Run-time Errors:** These occur while a program is running, after successful compilation, and cause abnormal termination or incorrect results. They happen due to invalid operations such as division by zero, invalid memory access, or file input/output failures.

**Types of run-time errors:**

- **Divide by Zero**

**Example:**



```c
#include <stdio.h>

int main() {
    int a = 10, b = 0;
    int result;

    result = a / b;    // ❌ division by zero
    printf("Result = %d", result);

    return 0;
}
```

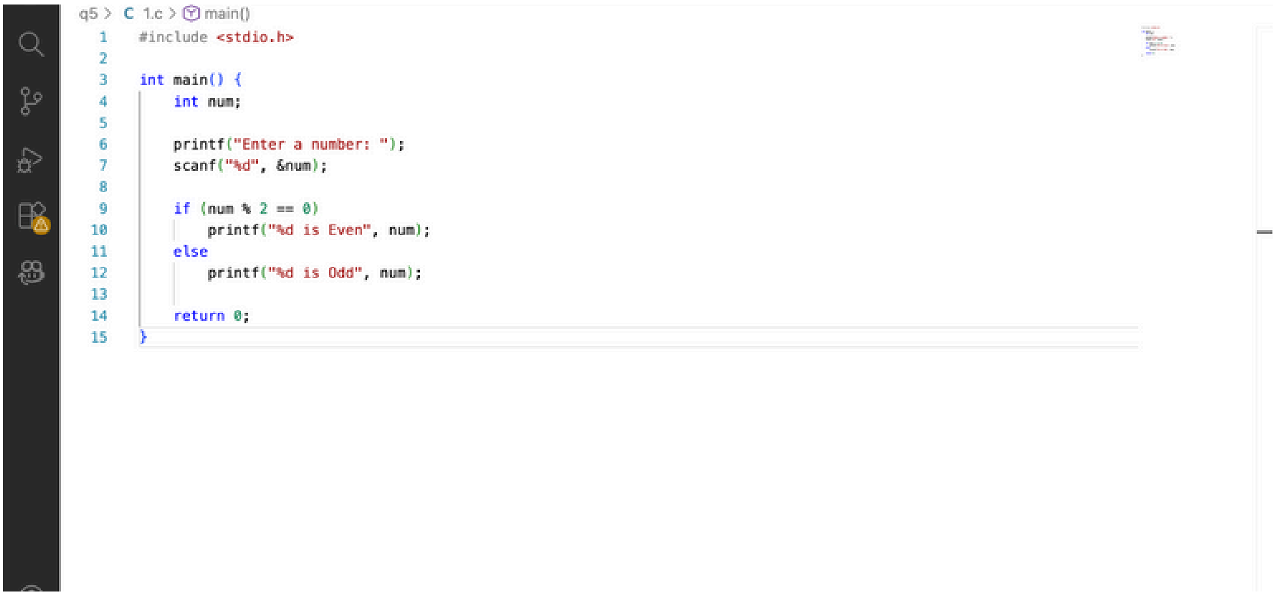
```
aryan@Aryans-MacBook-Air q4 % cd "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q4/" && gcc 1.c -o 1 && "/Users/aryan/Documents/
College/Practice/Practice-C/P1-5/q4/"1
Result = 0%
aryan@Aryans-MacBook-Air q4 % 
```

# Practical-5

**Aim: Odd/Even check using if-else, maximum of three numbers, ternary operator, and switch case.**

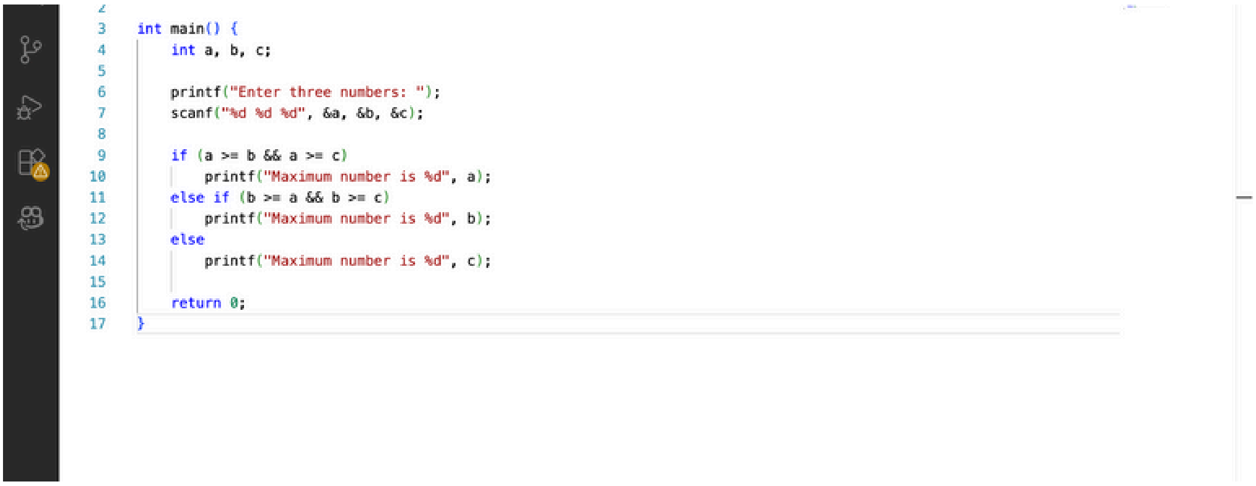
1. **If-Else (Definition):** The if-else statement is a decision-making control structure that allows a program to execute one block of code if a condition is true, and another block if the condition is false.

- Program to check if a number is even or odd using an if-else statement.



```c
#include <stdio.h>

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    if (num % 2 == 0)
        printf("%d is Even", num);
    else
        printf("%d is Odd", num);

    return 0;
}
```

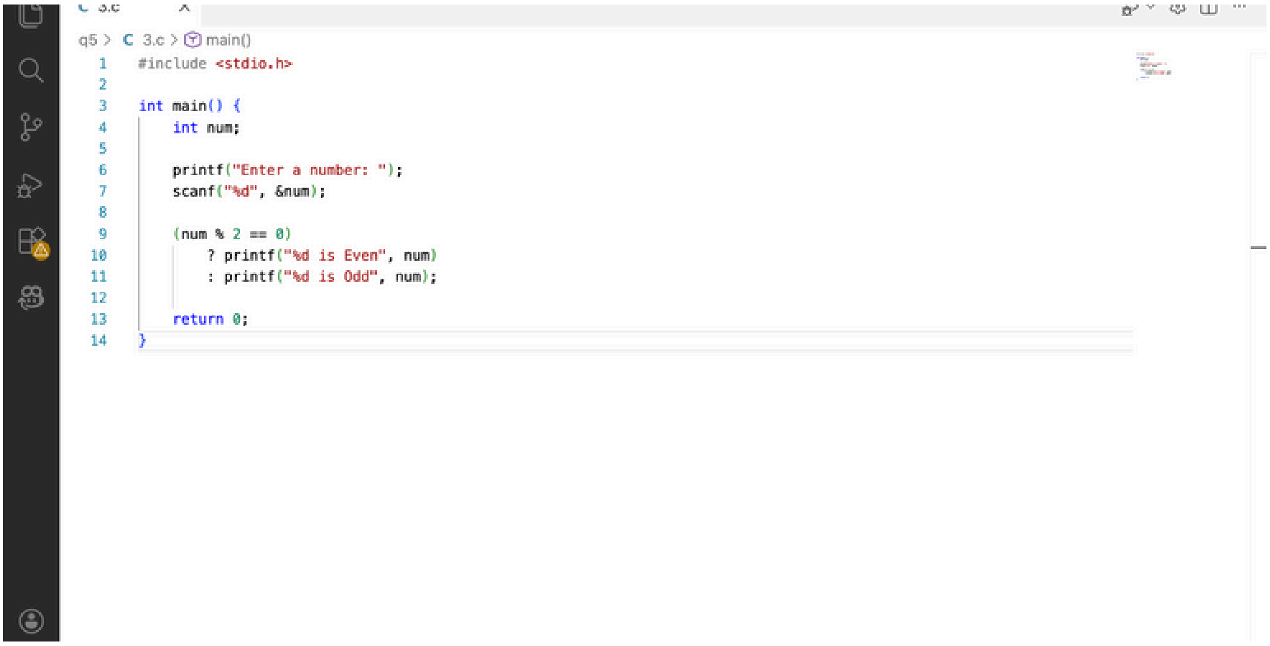
- Program to check the maximum of three numbers using an if-else statement.



```c
int main() {
    int a, b, c;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    if (a >= b && a >= c)
        printf("Maximum number is %d", a);
    else if (b >= a && b >= c)
        printf("Maximum number is %d", b);
    else
        printf("Maximum number is %d", c);

    return 0;
}
```

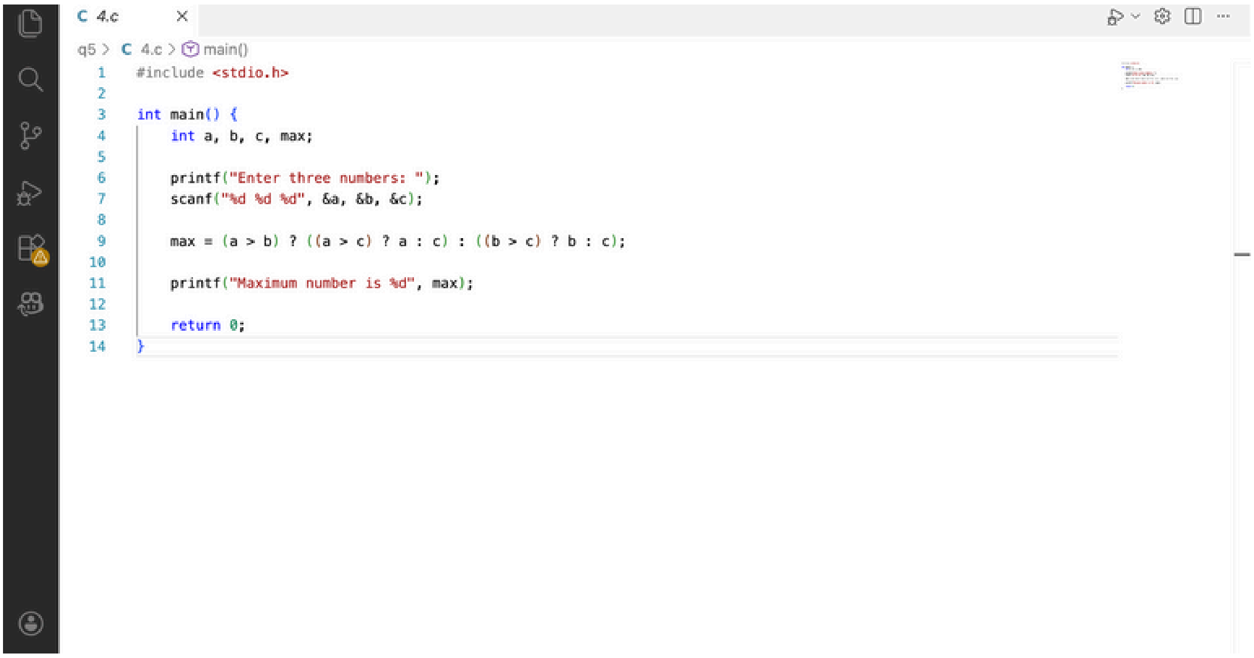
1. **Ternary operator:** The ternary operator in C, also known as the conditional operator, provides a concise way to express simple if-else statements in a single line.

- Program to check if a number is even or odd using the ternary operator.



```c
#include <stdio.h>

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    (num % 2 == 0)
        ? printf("%d is Even", num)
        : printf("%d is Odd", num);

    return 0;
}
```

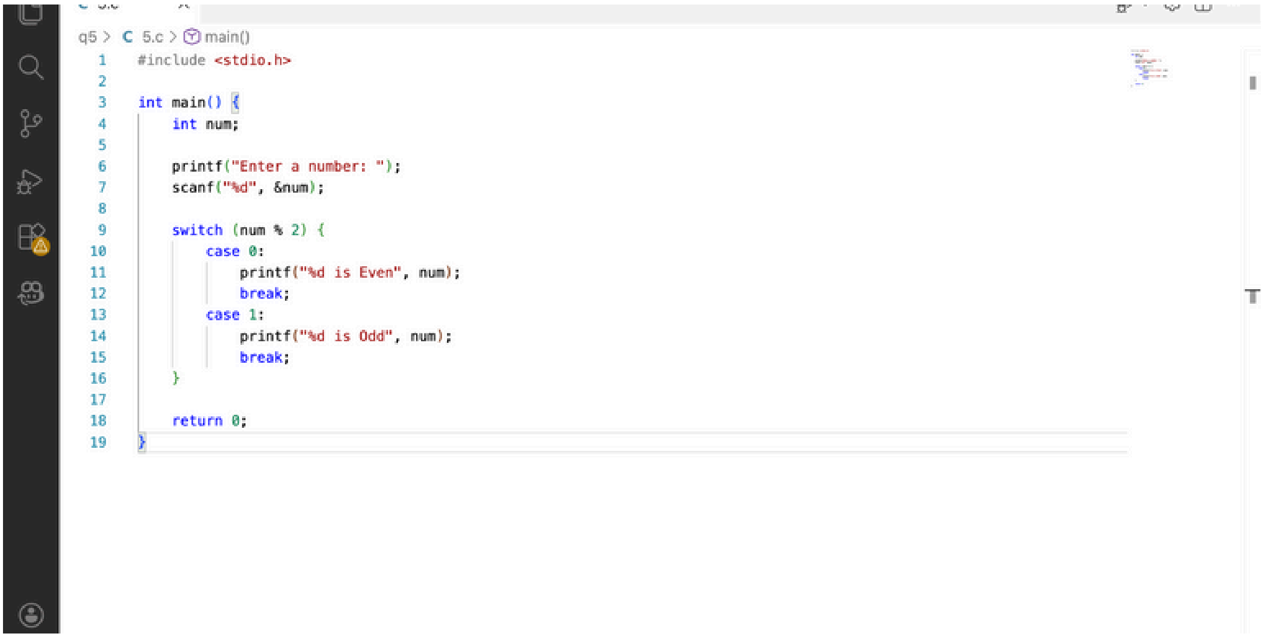
- Program to check the maximum of three numbers using the ternary operator.



```c
#include <stdio.h>

int main() {
    int a, b, c, max;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    max = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

    printf("Maximum number is %d", max);

    return 0;
}
```

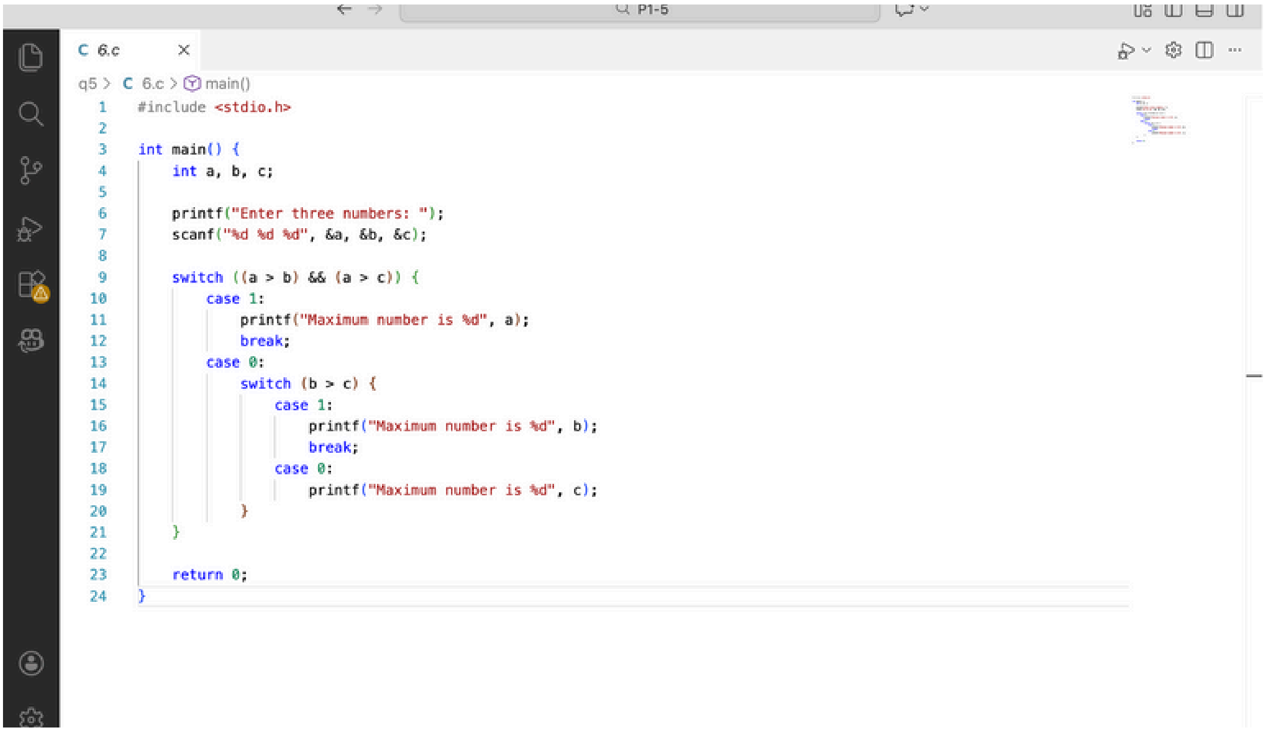
1. **Switch-Case:** The switch-case statement is a control structure used to execute one block of code from multiple choices based on the value of an expression.

- Program to check if a number is even or odd using the switch case.



```c
#include <stdio.h>

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    switch (num % 2) {
        case 0:
            printf("%d is Even", num);
            break;
        case 1:
            printf("%d is Odd", num);
            break;
    }

    return 0;
}
```

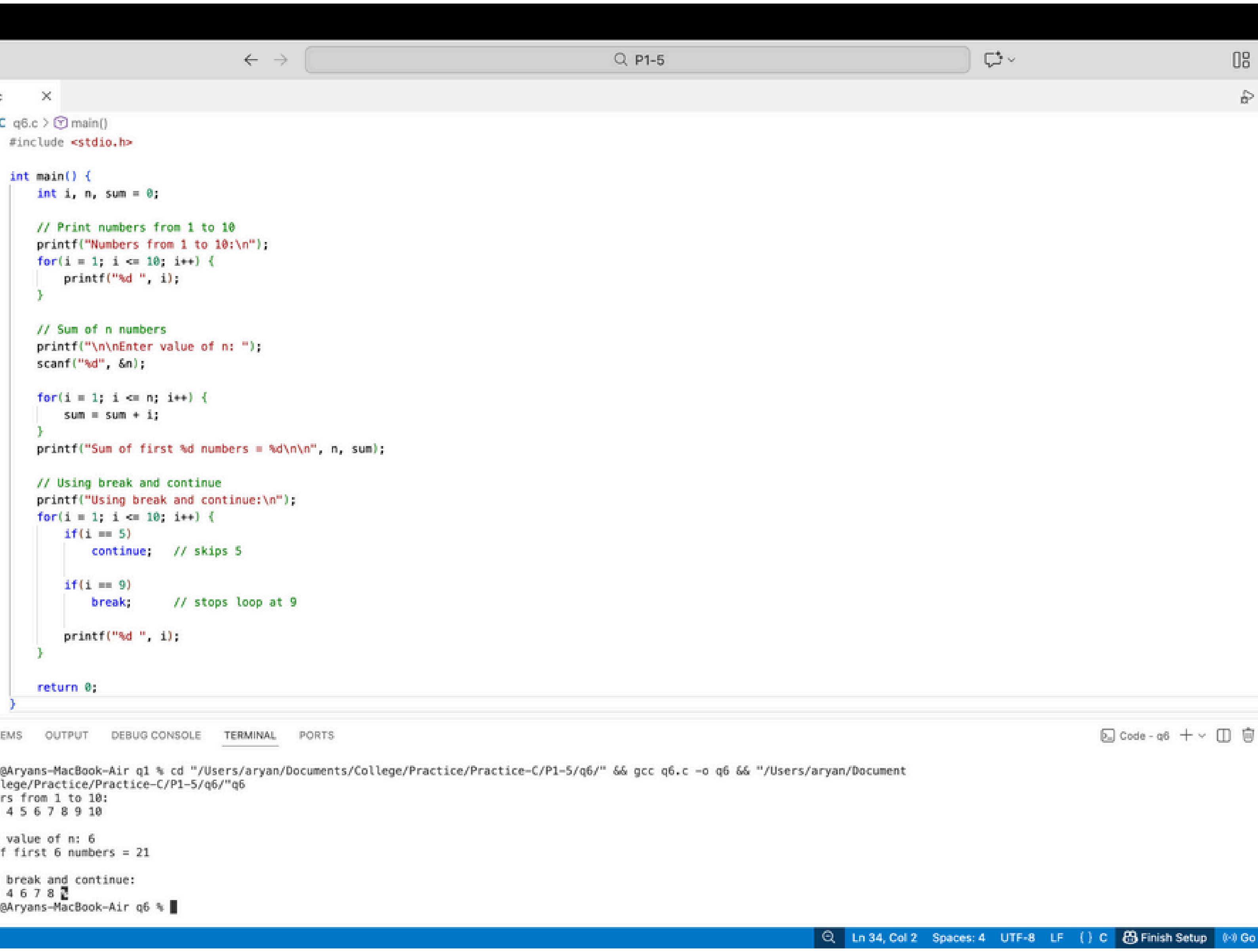
- Program to check the maximum of three numbers using the switch case.



```c
#include <stdio.h>

int main() {
    int a, b, c;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    switch ((a > b) && (a > c)) {
        case 1:
            printf("Maximum number is %d", a);
            break;
        case 0:
            switch (b > c) {
                case 1:
                    printf("Maximum number is %d", b);
                    break;
                case 0:
                    printf("Maximum number is %d", c);
            }
    }

    return 0;
}
```

# Practical-6

## Aim: Print numbers 1-10 using loop, sum of n numbers, use break & continue.



P1-5
q6.c > main()
EMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - q6
Ln 34, Col 2 Spaces: 4 UTF-8 LF { } C Finish Setup Go

```c
#include <stdio.h>

int main() {
    int i, n, sum = 0;

    // Print numbers from 1 to 10
    printf("Numbers from 1 to 10:\n");
    for(i = 1; i <= 10; i++) {
        printf("%d ", i);
    }

    // Sum of n numbers
    printf("\n\nEnter value of n: ");
    scanf("%d", &n);

    for(i = 1; i <= n; i++) {
        sum = sum + i;
    }
    printf("Sum of first %d numbers = %d\n\n", n, sum);

    // Using break and continue
    printf("Using break and continue:\n");
    for(i = 1; i <= 10; i++) {
        if(i == 5)
            continue;   // skips 5

        if(i == 9)
            break;      // stops loop at 9

        printf("%d ", i);
    }

    return 0;
}
```

```
@Aryans-MacBook-Air q1 % cd "/Users/aryan/Documents/College/Practice/Practice-C/P1-5/q6/" && gcc q6.c -o q6 && "/Users/aryan/Document
lege/Practice/Practice-C/P1-5/q6/"q6
rs from 1 to 10:
 4 5 6 7 8 9 10

 value of n: 6
f first 6 numbers = 21

 break and continue:
 4 6 7 8 %
@Aryans-MacBook-Air q6 % 
```