C Programming

Kriti Ghimire

Westcliff University

Fundamentals of C Programming

Nirajan Thakuri

February 14, 2025

Q1. What is the purpose of the main() function in a C program? Explain its significance.

The main() function is the starting point of every C program. When the program runs, execution begins from main(). It controls the overall flow by calling other functions and executing statements. The function usually returns an integer value to indicate whether the program ran successfully or encountered an error. It acts as the central hub of the program where all major tasks are organized and managed. The structure of the main() defines how the program will operate step by step. Without main(), a C program cannot execute. Overall, the main() plays a vital role in linking all parts of the program together and ensuring proper execution.

Q2. Explain the difference between a variable declaration and a variable initialization in C.

In C, a variable declaration means introducing a variable to the program by specifying its data type and name. It tells the compiler to reserve memory for that variable. However, at this stage, the variable does not have a defined value. For example, int x; declares a variable named x.

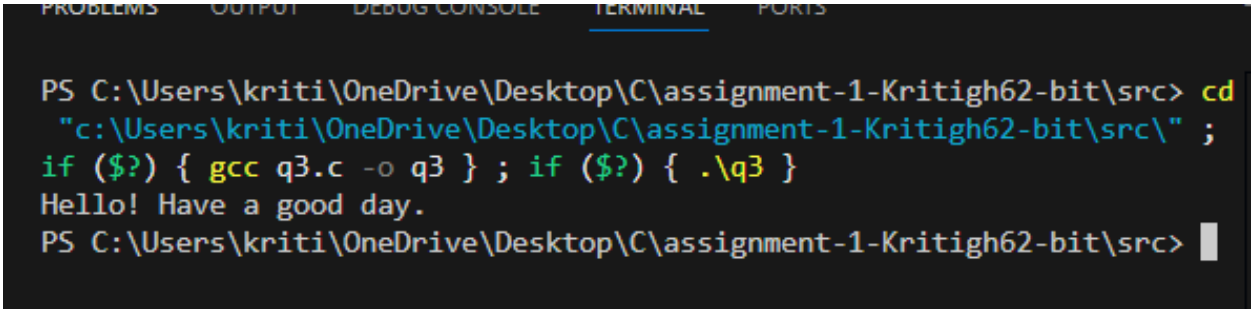
A variable initialization, on the other hand, means assigning a value to the variable when it is created. This gives the variable a definite starting value. For example, int x = 5; both declare the variable and initialize it with the value 5.

The key difference is that declaration only creates the variable and reserves memory, while initialization gives the variable its first meaningful value.

Q3. Write a C program to display a personalized greeting message. (Should contain 'hello' or 'welcome' in the message)

```c
#include <stdio.h>

int main() {
    printf("Hello! Have a good day.");
    return 0;
}
```

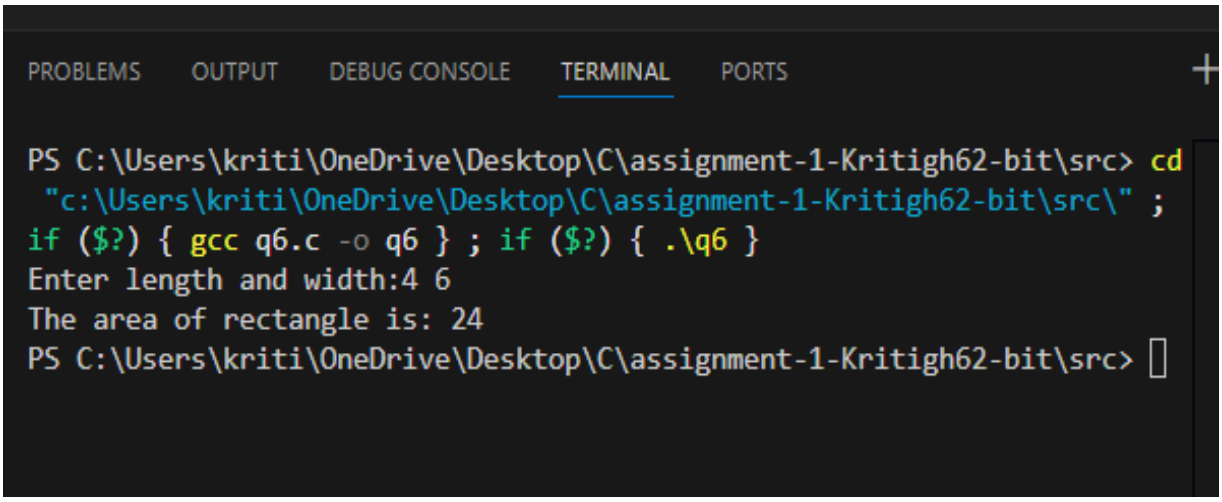
Output



Q6. Write a C program to calculate the area of a rectangle. Prompt the user to enter the length and width, and display the result.

```c
#include<stdio.h>
int main ()
{
    // Declare variables: l=length, w= width, a=area//
    int l, w, a;
    //prompt the user to enter length and width//
    printf("Enter length and width:", l, w);
    // Read length and width//
    scanf("%d %d", &l , &w);
    // Calculate the area of a rectangle//
    a = l * w ;
    // Display the calculated output//
    printf("The area of rectangle is %d", a);
    return 0;

}
```

Output



Q7. What is the role of the scanf() function in C? Provide an example of its usage.

In C, the scanf() function is the general-purpose input function, which is used to read input from the user and store it in variables during program execution. It reads data from the keyboard and requires format specifiers like %d for integers, %f for floats, and %c for characters to determine the input type. The & operator is used to pass the address of the variable so that scanf() can store the value at the correct memory location.

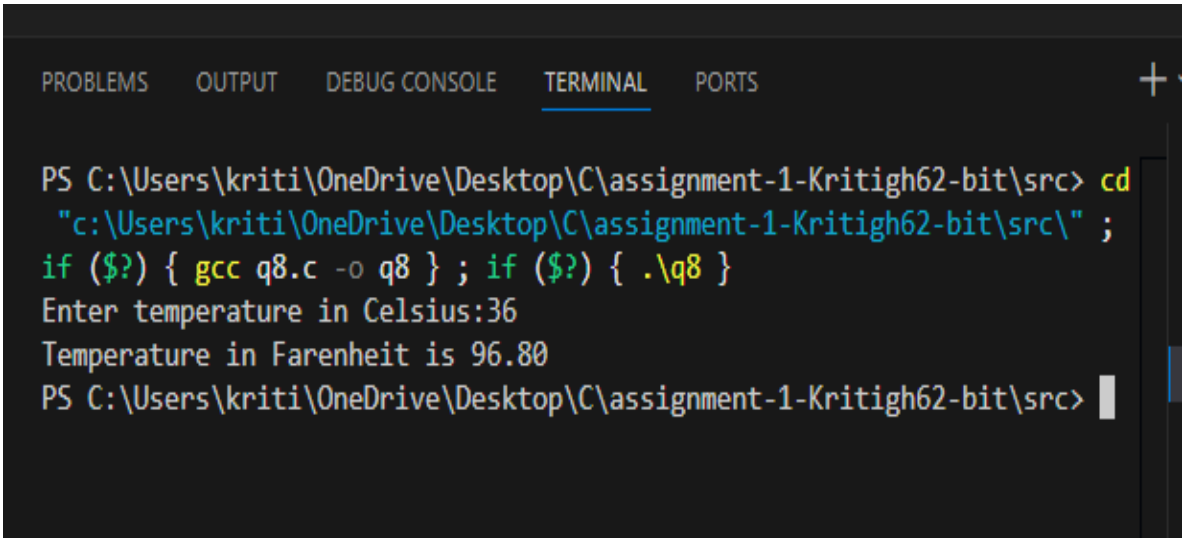
For example:

```
#include<stdio.h>
int main(){
    // Age, Height, Initial //
    int age;
    float height;
    char initial;
    printf("Enter age, height, initial:");
    scanf("%d %f %c", &age, &height, &initial);
    printf("Age: %d\nHeight: %f\nInitial: %c\n",age, height, initial);
    return 0;
}
```

Q8. Write a C program to convert temperature from Celsius to Fahrenheit. Prompt the user for a temperature in Celsius and display the equivalent temperature in Fahrenheit. (Formula: fahrenheit = (celsius * 9 / 5) + 32)

```c
#include<stdio.h>
int main()
{
    float Celsius, Fahrenheit;
    printf("Enter temperature in Celsius:", Celsius);
    scanf("%f", &Celsius);

    Fahrenheit = (Celsius * 9 / 5) + 32;

    printf("Temperature in Farenheit is %.2f\n",farenheit );
    return 0;
}
```

Output



Q9. Input a number representing days and print the equivalent number of weeks and days. (Example: 10 days = 1 week and 3 days)

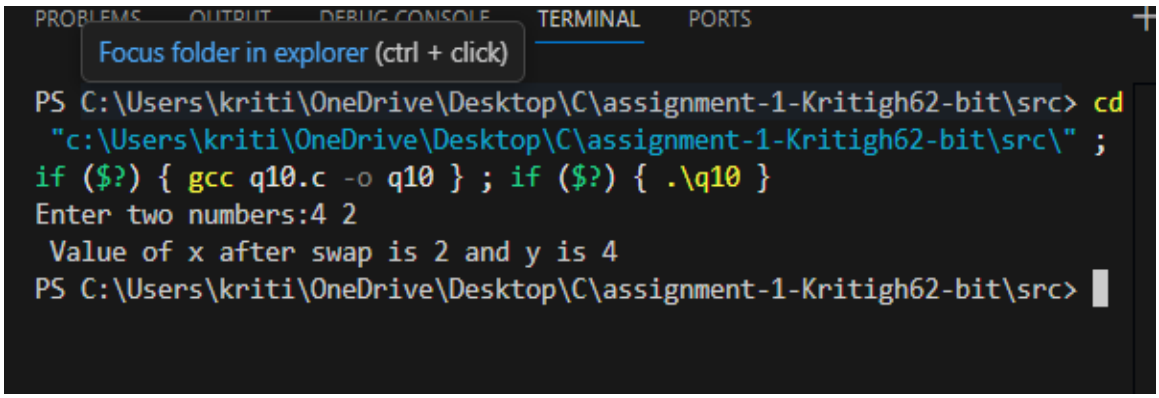
```c
#include<stdio.h>
int main()
{
    int days, weeks;
    printf("Enter a number:");
    scanf("%d", &days);
    weeks = days / 7;
    days = days % 7;
     printf("%d weeks and %d days", weeks , days);
     return 0;

}
```

Output

Q10. Write a C program to swap the values of two variables using a temporary variable.

```c
#include<stdio.h>
int main()
{
    int x,y;
    int temp;
    printf("Enter two numbers:");
    scanf("%d %d", &x, &y);
    temp = x;
    x = y;
    y = temp;
    printf(" Value of x after swap is %d and y is %d\n", x, y);
    return 0;
}
```

Output



Q11. Write a C expression that performs the following operations in a single line: increment a variable by 1, multiply it by 3, and subtract 10.

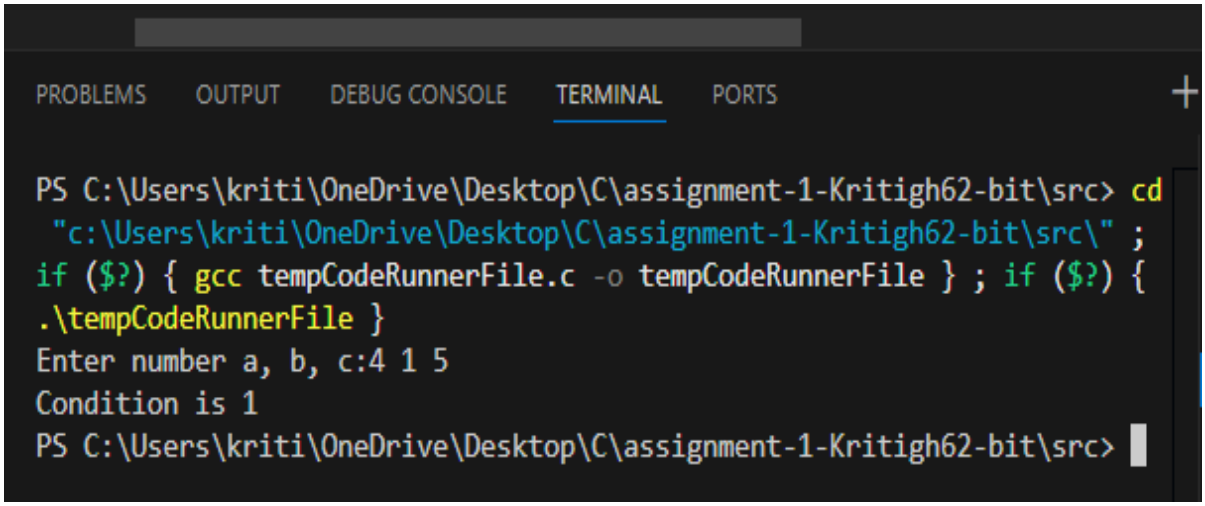
```c
#include<stdio.h>
int main()
{
    int x;
    printf("Enter a number:");
    scanf("%d", &x);

    x = (x + 1) * 3 - 10;
    printf("Result: %d\n", x);
    return 0;
}
```

Output

Q12. Given three variables a, b, and c, write an expression that checks if a is greater than b and c is not equal to 0.

```c
#include <stdio.h>
int main()
{
    int a, b, c;
    printf("Enter number a, b, c:");
    scanf("%d %d %d", &a, &b, &c);

    int result;
    result = a > b && c != 0;

    printf("Condition is %d", result);
    // if (a > b && c != 0) {
    //     printf("%d is greater than %d", a, b);
    // } else {
    //     printf("%d is not equal to 0", c);
    // }

    return 0;
}
```

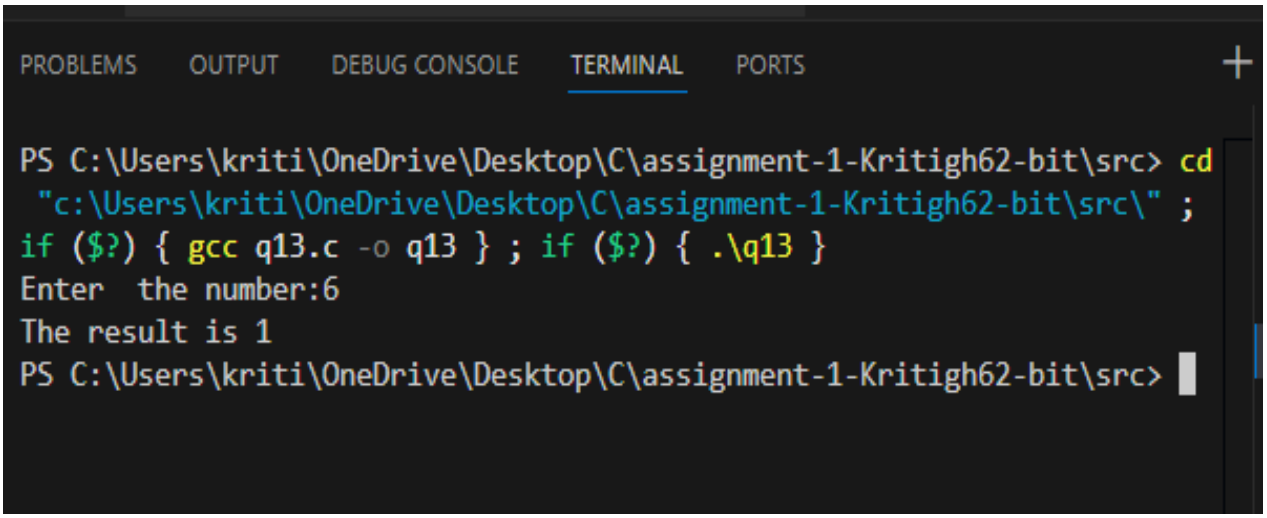
Output



Q13. Write a C expression that evaluates whether a number is divisible by both 2 and 3 (without using the modulus operator).

```c
#include<stdio.h>

int main()
{
    int num;
    printf("Enter  the number:");
    scanf("%d", &num);
    int result;
    result = (num % 2 == 0) && (num % 3 == 0);
    printf("The result is %d\n", result );
    return 0;



}
```

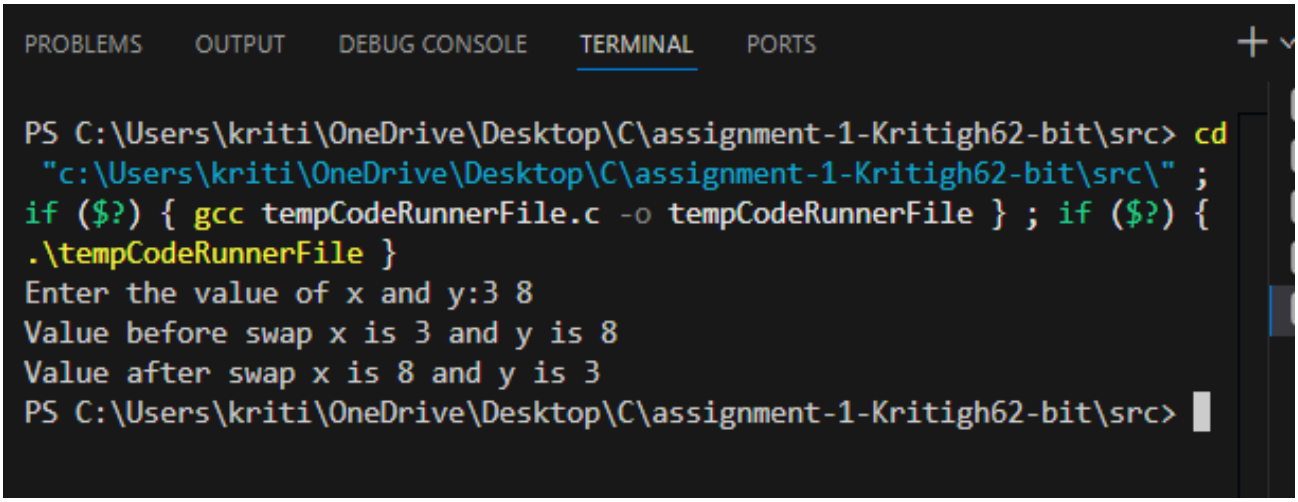
Output



Q14. Create an expression that swaps the values of two variables, x and y without using a temporary variable.

```c
#include<stdio.h>
int main()
{
    int x , y;
    printf("Enter the value of x and y:");
    scanf("%d %d", &x, &y);
    printf("Value before swap x is %d and y is %d\n", x , y);
    x = x + y;
    y = x - y;
    x = x - y;
    printf("Value after swap x is %d and y is %d\n", x , y);
    return 0;
}
```

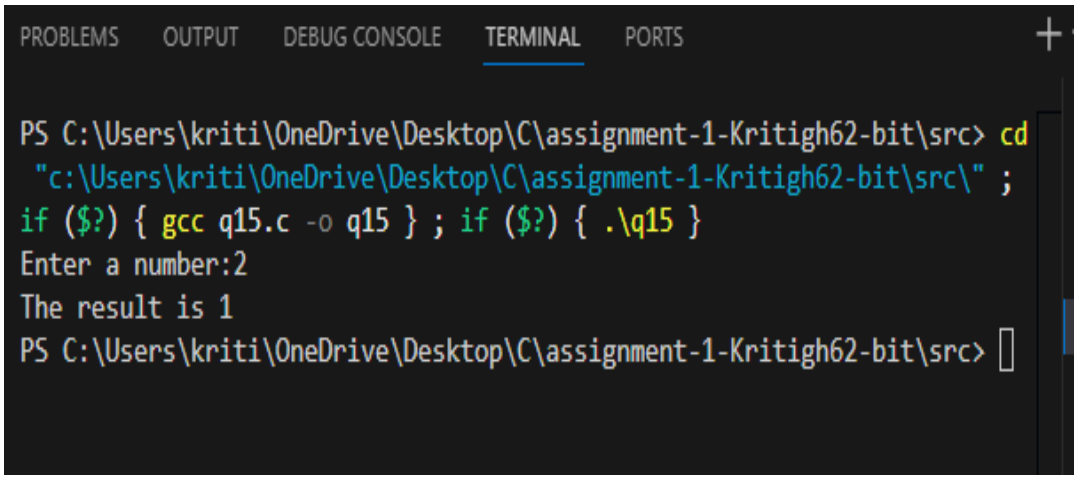
Output



Q15. Write an expression that checks if a number is both positive and even.

```c
#include<stdio.h>
int main()
{
   int num;
    printf("Enter a number:");
    scanf("%d", &num);

    int result;
    result =  num > 0 && num % 2 == 0 ;
    printf("The result is %d\n", result);


    return 0;
}
```

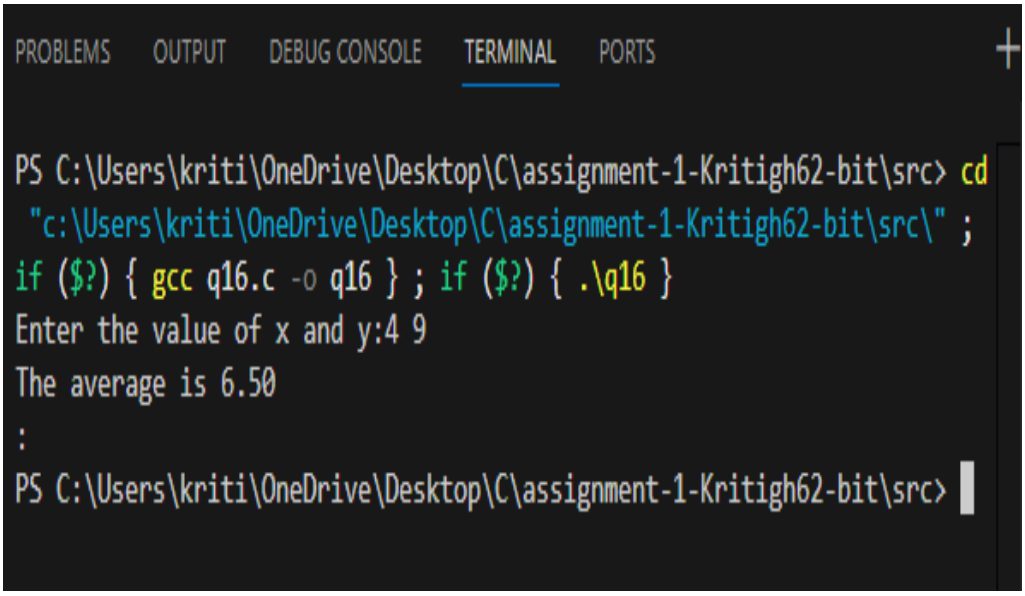
Output



Q16. Given two variables x and y, write an expression that calculates the average of their values.

```c
#include<stdio.h>
int main()
{
    float x,y,average;
    printf("Enter the value of x and y:");
    scanf("%f %f", &x, &y);
    average = (x+y)/2;
    printf("The average is %.2f\n:", average);
    return 0;

}
```

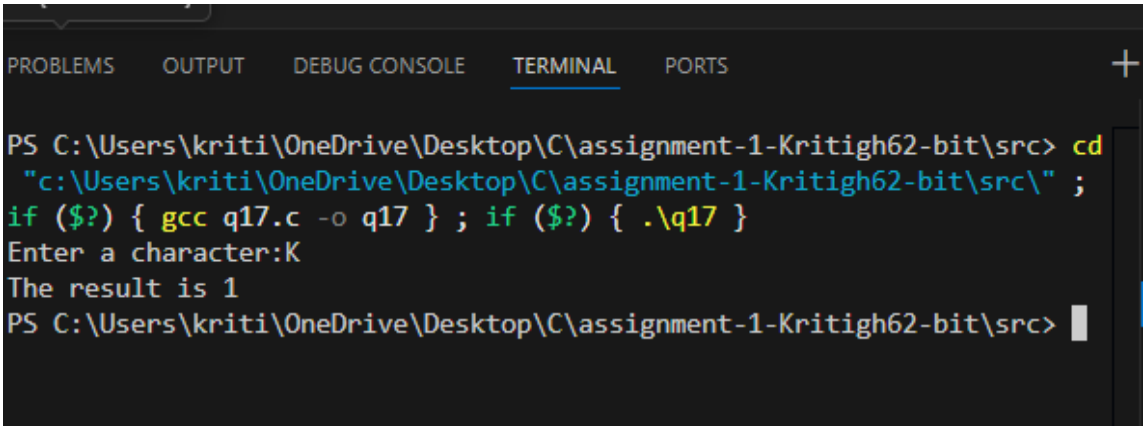
Output



Q17. Create an expression that checks if a given character is an uppercase letter.

```c
#include<stdio.h>
int main()
{
    char character;
    int result;
    printf("Enter a character:");
    scanf("%c", &character);
    result = (character  >= 'A') && (character <= 'Z');
    printf("The result is %d\n", result);
    return 0;

}
```

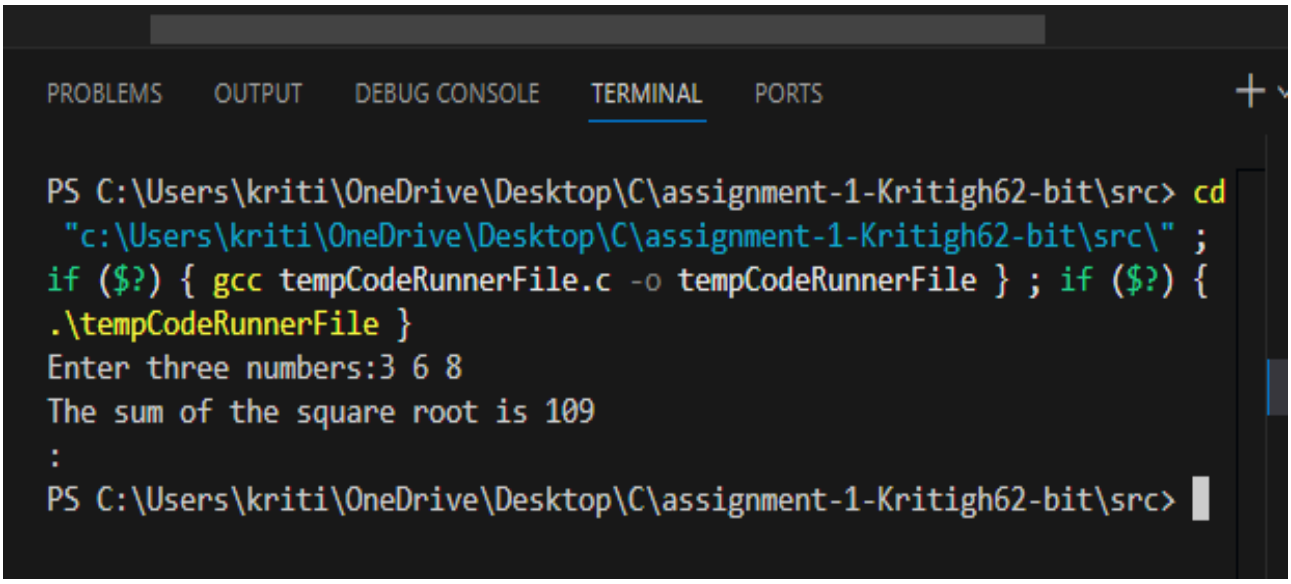
Output



Q18. Write a C expression that calculates the sum of the squares of three different numbers.

```c
#include<stdio.h>
int main()
{
    int a, b, c, sum;
    printf("Enter three numbers:");
    scanf("%d %d %d", &a, &b, &c);
     sum = (a*a) + (b*b) + (c*c);
     printf("The sum of the square root is %d\n:", sum);
     return 0;


}
```

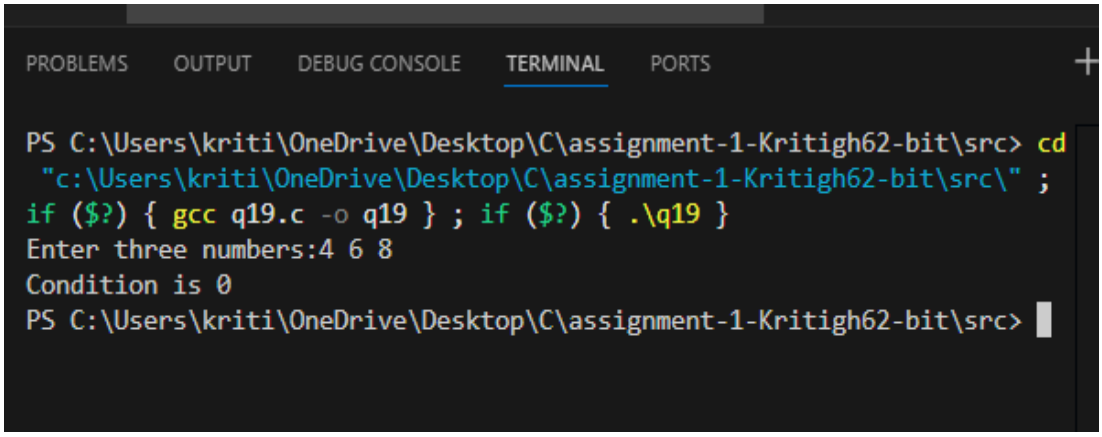
Output



Q19. Given three variables a, b, and c, write an expression that checks if a is equal to b and b is not equal to c.

```c
#include<stdio.h>
int main()
{
    int a, b, c;
    printf("Enter three numbers:");
    scanf("%d %d %d", &a, &b, &c);

    int result;
    result = a == b && b != c;
        printf("Condition is %d", result);


    return 0;

}
```

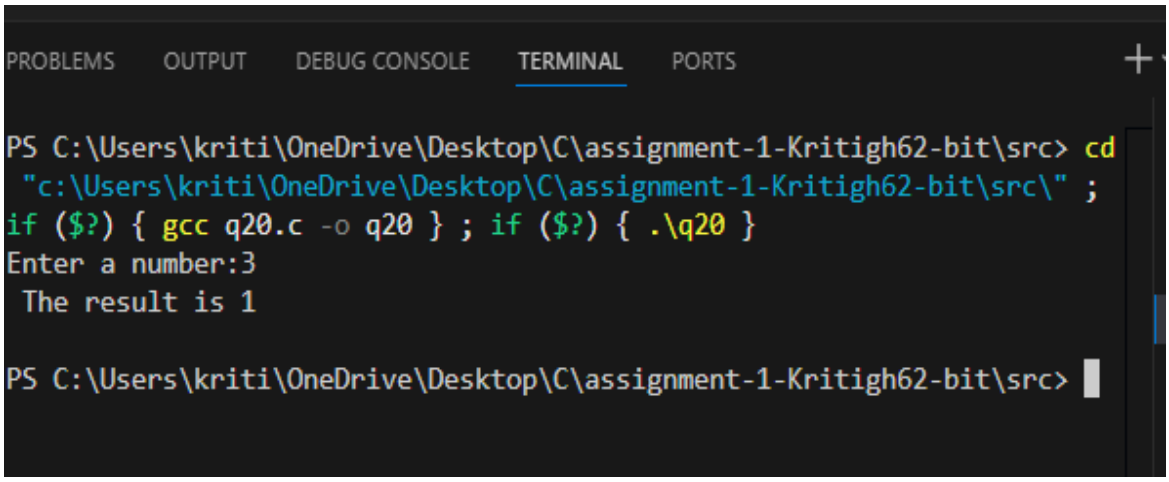
Output



Q20. Write an expression that checks if a number is a multiple of either 3 or 5.

```c
#include<stdio.h>
int main()
{
  int x;
    printf("Enter a number:");
    scanf("%d", &x);

    int result;
    result = x % 3==0 || x % 5==0 ;
        printf(" The result is %d\n ", result);

return 0;
}
```

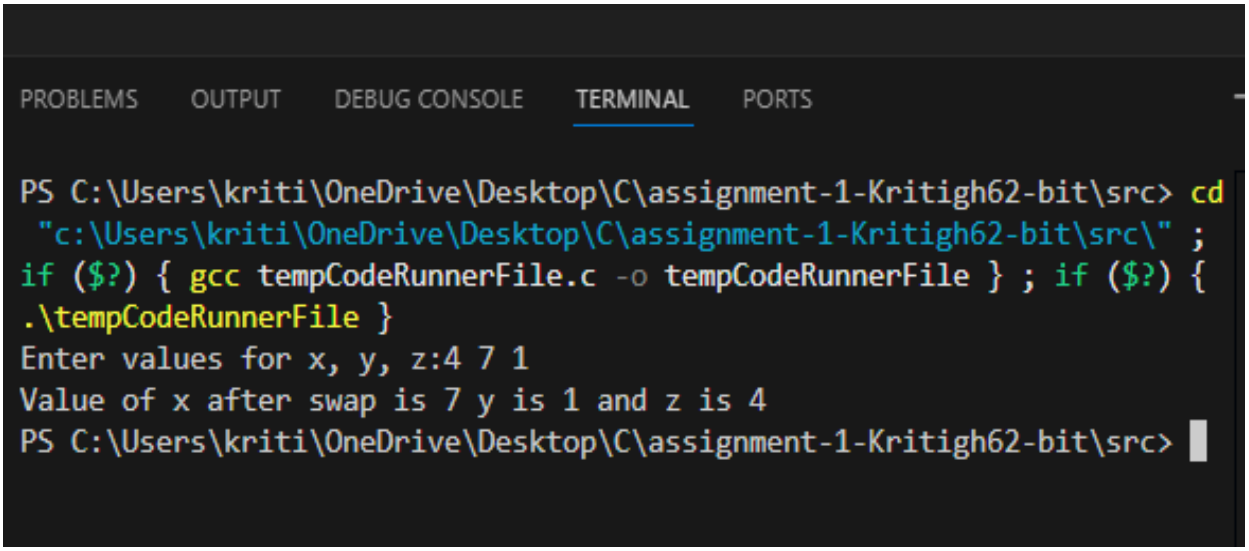
Output



Q21. Create an expression that swaps the values of three variables x, y, and z in a cyclic order (i.e., x becomes y, y becomes z, and z becomes x).

```c
#include<stdio.h>
int main()
{
    int x, y, z, temp;
    printf("Enter values for x, y, z:");
    scanf("%d %d %d",&x, &y, &z );

    temp = x;
    x = y;
    y = z;
    z = temp;

    printf("Value of x after swap is %d y is %d and z is %d\n", x, y, z);
    return 0;
}
```

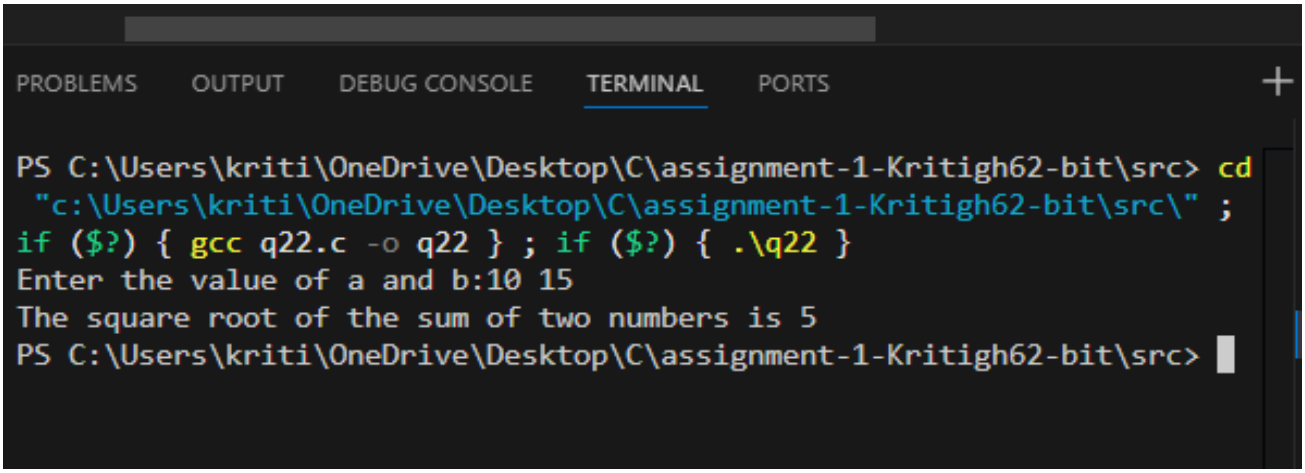
Output



Q22. Write a C expression that calculates the square root of the sum of two numbers, rounded to the nearest integer.

```c
#include<stdio.h>
#include<math.h>
int main()
{
    int a, b, sum;
    printf("Enter the value of a and b:");
    scanf("%d %d", &a, &b);
    sum = (a + b);
    int result;
    result = sqrt(sum);
    printf("The square root of the sum of two numbers is %d\n", result);
    return 0;
}
```

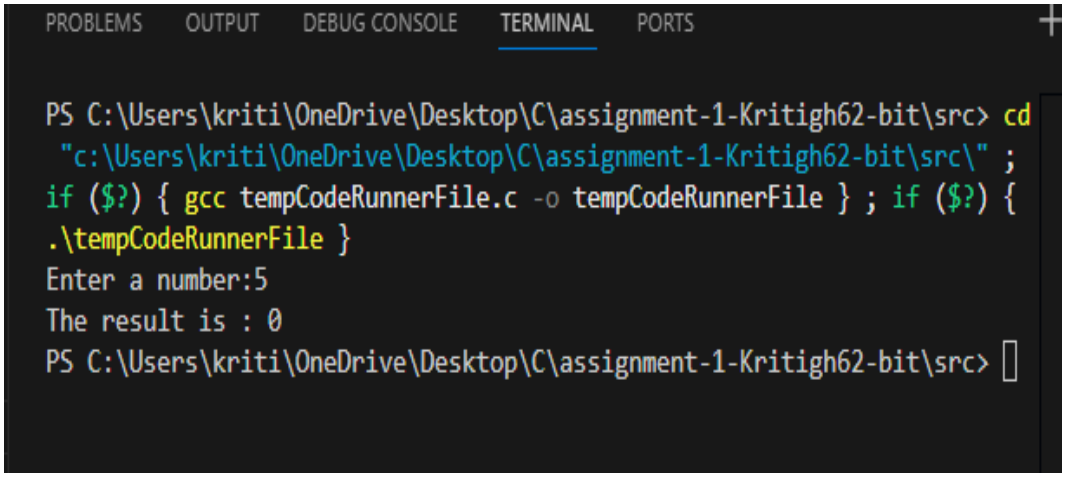
Output



Q23. Given a variable num, write an expression that checks if it is a power of 2.

```c
#include<stdio.h>
int main()
{
    int num, result;
    printf("Enter a number:");
    scanf("%d", &num);

    result = !(num & (num - 1));
    printf("The result is : %d\n", result);
    return 0;
}
```

Output



Q24. Create an expression that checks if a given number is a perfect square.

```c
#include<stdio.h>
#include<math.h>
int main()
{
    int num;
    printf("Enter a number:");
    scanf("%d", &num);
    int x = sqrt(num);
    int result = (x * x == num);
    printf("The result is : %d\n", result);
    return 0;
}
```

Output