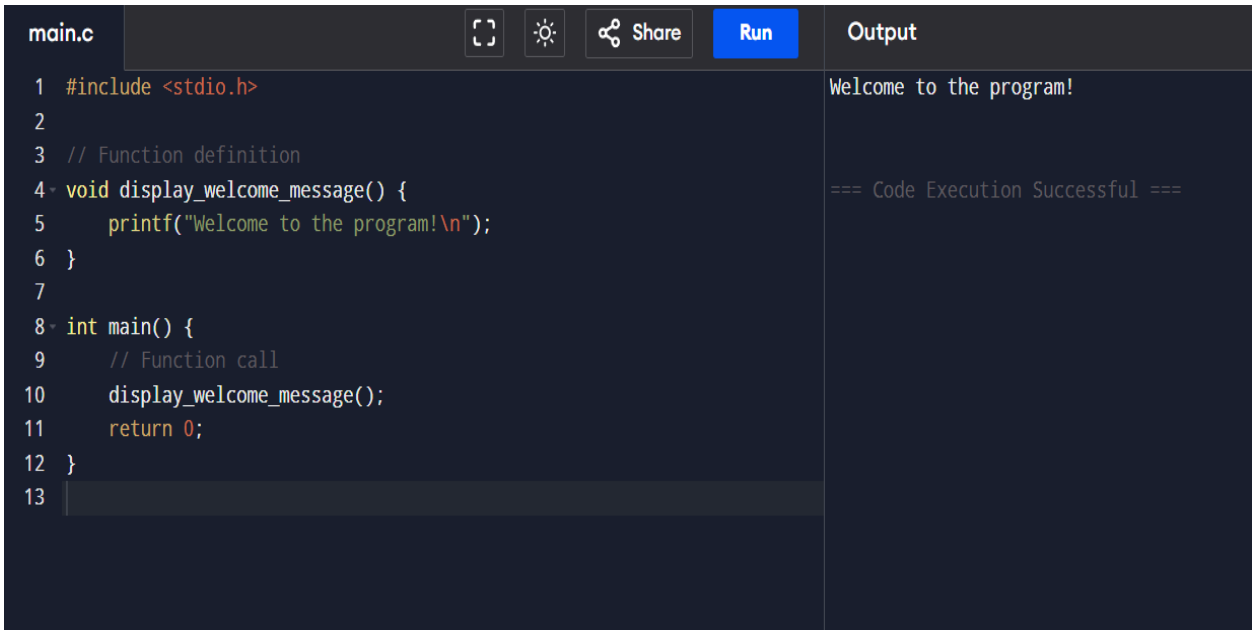
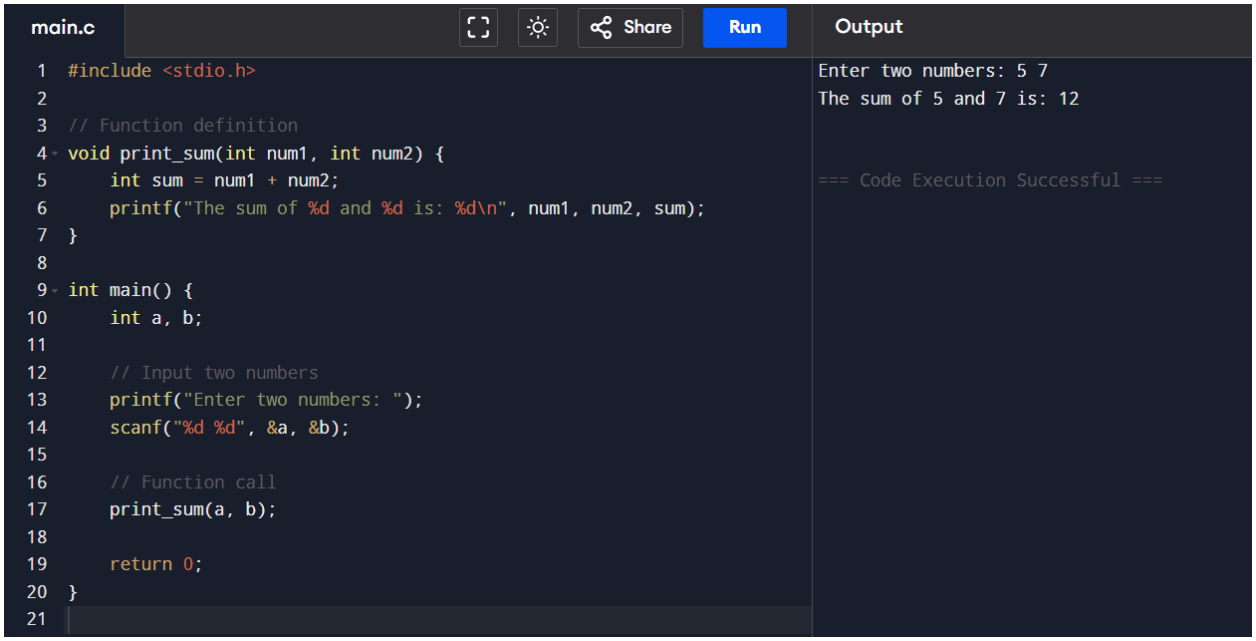
DAY-3 TASK(TEAM-D)

1. Write a program to display a welcome message using a function without parameters and return value:



2. Write a program where a function accepts two numbers as arguments and prints their sum. Use a function with parameters and no return value:



3.Write a program where a function takes two integers as parameters, compares them, and returns the maximum value:

main.c — Share — Run — Output

```c
#include <stdio.h>

// Function definition
int find_max(int num1, int num2) {
    if (num1 > num2) {
        return num1;  // Return num1 if it is greater
    } else {
        return num2;  // Return num2 if it is greater or equal
    }
}

int main() {
    int a, b;

    // Input two numbers
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    // Call the function and print the result
    int max_value = find_max(a, b);
    printf("The maximum value is: %d\n", max_value);

    return 0;
}
```

```
Enter two numbers: 10 20
The maximum value is: 20


=== Code Execution Successful ===
```

4.Write a program to compute the factorial of a number using a recursive function:

main.c — Share — Run — Output

```c
#include <stdio.h>

// Recursive function to calculate factorial
int factorial(int n) {
    if (n == 0 || n == 1) {
        return 1;  // Base case: 0! = 1 and 1! = 1
    } else {
        return n * factorial(n - 1);  // Recursive case: n! = n * (n
            -1)!
    }
}

int main() {
    int number;

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

    // Check for negative input
    if (number < 0) {
        printf("Factorial is not defined for negative numbers.\n");
    } else {
        // Calculate and print the factorial
        int result = factorial(number);
```

```
Enter a number: 5
The factorial of 5 is: 120


=== Code Execution Successful ===
```

5.Write a program to demonstrate swapping two numbers using a function with call by value:

main.c — Share — Run — Output

```c
#include <stdio.h>

// Function to swap two numbers using call by value
void swap(int a, int b) {
    int temp;

    // Swap logic
    temp = a;
    a = b;
    b = temp;

    // Print the swapped values inside the function
    printf("Inside swap function: a = %d, b = %d\n", a, b);
}

int main() {
    int x, y;

    // Input two numbers
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);

    // Print original values before swapping
    printf("Before swap: x = %d, y = %d\n", x, y);

```

```
Enter two numbers: 10 20
Before swap: x = 10, y = 20
Inside swap function: a = 20, b = 10
After swap (in main): x = 10, y = 20


=== Code Execution Successful ===
```

6.Write a program where a function accepts an array and its size as arguments and returns the largest value in the array:

main.c — Share — Run — Output

```c
#include <stdio.h>

// Function to find the largest element in an array
int find_largest(int arr[], int size) {
    int largest = arr[0]; // Assume the first element is the largest

    // Loop through the array to find the largest element
    for (int i = 1; i < size; i++) {
        if (arr[i] > largest) {
            largest = arr[i];
        }
    }

    return largest; // Return the largest value
}

int main() {
    int n;

    // Input the size of the array
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n]; // Declare the array of size n
```

```
Enter the number of elements: 5
Enter the elements of the array: 5 6 7 8 9
The largest element in the array is: 9


=== Code Execution Successful ===
```

7.Write a program in C to print all perfect numbers in given range using the function:

main.c — Share — Run

```c

// Function to print perfect numbers in a given range
void print_perfect_numbers(int start, int end) {
    printf("Perfect numbers in the range %d to %d are: \n", start,
        end);
    for (int i = start; i <= end; i++) {
        if (is_perfect(i)) {
            printf("%d ", i);
        }
    }
    printf("\n");
}

int main() {
    int start, end;

    // Input the range
    printf("Enter the range (start and end): ");
    scanf("%d %d", &start, &end);

    // Call the function to print perfect numbers in the range
    print_perfect_numbers(start, end);

    return 0;
}
```

Output

```
Enter the range (start and end): 1 1000
Perfect numbers in the range 1 to 1000 are:
6 28 496


=== Code Execution Successful ===
```

8.Write a program to reverse a number using function?(Get the input from user):

main.c — Share — Run

```c
#include <stdio.h>

// Function to reverse a number
int reverse_number(int num) {
    int reversed = 0;

    while (num != 0) {
        int digit = num % 10;           // Get the last digit of the
            number
        reversed = reversed * 10 + digit;  // Append the digit to
            reversed number
        num /= 10;                      // Remove the last digit
            from the number
    }

    return reversed; // Return the reversed number
}

int main() {
    int number;

    // Input the number from user
    printf("Enter a number: ");
    scanf("%d", &number);
```

Output

```
Enter a number: 12345
The reversed number is: 54321


=== Code Execution Successful ===
```

9.Write a menu-driven program where each arithmetic operation is implemented using a separate function:

main.c

Share Run

```c
#include <stdio.h>

// Function to add two numbers
float add(float num1, float num2) {
    return num1 + num2;
}

// Function to subtract two numbers
float subtract(float num1, float num2) {
    return num1 - num2;
}

// Function to multiply two numbers
float multiply(float num1, float num2) {
    return num1 * num2;
}

// Function to divide two numbers
float divide(float num1, float num2) {
    if (num2 == 0) {
        printf("Error! Division by zero is not allowed.\n");
        return 0;  // Return 0 in case of division by zero
    }
    return num1 / num2;
}
```

Output

```
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 3
Enter two numbers: 2 5
The product of 2.00 and 5.00 is: 10.00

Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 1
Enter two numbers: 2 3
The sum of 2.00 and 3.00 is: 5.00

Menu:
1. Addition
2. Subtraction
3. Multiplication
```