1- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int findSumResult = findSum(5, 7);
    std::cout << "Sum: " << findSumResult << "\n";

    double calculatePowerResult = calculatePower(2.0, 3);
    std::cout << "Power: " << calculatePowerResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>
#include <cmath>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

int findSum(int a, int b) {
    return a + b;
}

double calculatePower(double base, int exponent) {
    return std::pow(base, exponent);
}
```

Solution

```
Hello, welcome to the program!
Sum: 12
Power: 8
```

2- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int countDigitsResult = countDigits(12345);
    std::cout << "Number of Digits: " << countDigitsResult << "\n";

    bool isPalindromeResult = isPalindrome("level");
    std::cout << "Is Palindrome? " << std::boolalpha << isPalindromeResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>
#include <cstring>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

int countDigits(int number) {
    int count = 0;
    while (number != 0) {
        number /= 10;
        ++count;
    }
    return count;
}

bool isPalindrome(const char* str) {
    int length = std::strlen(str);
    for (int i = 0; i < length / 2; ++i) {
        if (str[i] != str[length - 1 - i]) {
            return false;
        }
    }
    return true;
}
```

Solution

```
Hello, welcome to the program!
Number of Digits: 5
Is Palindrome? true
```

3- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int multiplyResult = multiplyNumbers(5, 6);
    std::cout << "Product: " << multiplyResult << "\n";

    bool isLeapYearResult = isLeapYear(2024);
    std::cout << "Is Leap Year? " << std::boolalpha << isLeapYearResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

int multiplyNumbers(int a, int b) {
    return a * b;
}

bool isLeapYear(int year) {
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
        return true;
    } else {
        return false;
    }
}
```

Solution

```
Hello, welcome to the program!
Product: 30
Is Leap Year? true
```

4- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int factorialResult = calculateFactorial(5);
    std::cout << "Factorial: " << factorialResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

int calculateFactorial(int n) {
    int num = 1;
    for(int i = 1 ; i <= n; i++ )
       num*=i;
  return num;
}
```

Solution

```
Hello, welcome to the program!
Factorial: 120
```

5- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int reverseResult = reverseNumber(12345);
    std::cout << "Reversed Number: " << reverseResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

int reverseNumber(int number) {
    int reversed = 0;
    while (number != 0) {
        reversed = reversed * 10 + number % 10;
        number /= 10;
    }
    return reversed;
}
```

Solution

```
Hello, welcome to the program!
Reversed Number: 54321
```

6- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int sumOfDigitsResult = sumOfDigits(123);
    std::cout << "Sum of Digits: " << sumOfDigitsResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

int sumOfDigits(int number) {
    int sum = 0;
    while (number != 0) {
        sum += number % 10;
        number /= 10;
    }
    return sum;
}
```

Solution

```
Hello, welcome to the program!
Sum of Digits: 6
```

7- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    bool isPerfectSquareResult = isPerfectSquare(16);
    std::cout << "Is Perfect Square? " << std::boolalpha << isPerfectSquareResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>
#include <cmath>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

bool isPerfectSquare(int number) {
    int squareRoot = static_cast<int>(std::sqrt(number));
    return (squareRoot * squareRoot == number);
}
```

Solution

```
Hello, welcome to the program!
Is Perfect Square? true
```

8- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int gcdResult = calculateGCD(24, 36);
    std::cout << "GCD: " << gcdResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

int calculateGCD(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}
```

Solution

```
Hello, welcome to the program!
GCD: 12
```

9- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int lcmResult = calculateLCM(8, 12);
    std::cout << "LCM: " << lcmResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

int calculateLCM(int a, int b) {
    int max = (a > b) ? a : b;
    int lcm = max;

    while (true) {
        if (lcm % a == 0 && lcm % b == 0) {
            break;
        }
        lcm += max;
    }

    return lcm;
}
```

Solution

```
Hello, welcome to the program!
LCM: 24
```

10- Trace the following program and predict the output.

```cpp
// www.gammal.tech

// main.cpp
#include <iostream>
#include "functions.h"

int main() {
    printMessage();

    int countPrimesResult = countPrimesInRange(10, 50);
    std::cout << "Number of Primes: " << countPrimesResult << "\n";

    return 0;
}
```

```cpp
// www.gammal.tech

// functions.cpp
#include <iostream>

void printMessage() {
    std::cout << "Hello, welcome to the program!\n";
}

bool isPrime(int number) {
    if (number < 2) {
        return false;
    }
    for (int i = 2; i * i <= number; ++i) {
        if (number % i == 0) {
            return false;
        }
    }
    return true;
}

int countPrimesInRange(int start, int end) {
    int count = 0;
    for (int i = start; i <= end; ++i) {
        if (isPrime(i)) {
            ++count;
        }
    }
    return count;
}
```

Solution

```
Hello, welcome to the program!
Number of Primes: 11
```