1.Explain the difference between int, float, and char with examples.

🔹 `int` – Integer

Definition:
Stores whole numbers, both positive and negative, using binary representation.

Key Characteristics:

- Typically occupies **4 bytes (32 bits)** in modern systems.
- Range for a 32-bit signed int:
  −231-2^{31}−231 to 231−12^{31} - 1231−1 (i.e., from -2,147,483,648 to 2,147,483,647)
- Stored in **two's complement** form for negative numbers.

Use Case Example (C language):

c
CopyEdit

```
int speed = 90;
```

Binary Example:

- `int x = 5;` → `00000000 00000000 00000000 00000101`
- `int y = -5;` → `11111111 11111111 11111111 11111011` (two's complement)

🔹 `float` – Floating-Point Number

Definition:
Stores real numbers (with fractional part) using IEEE 754 standard.

Key Characteristics:

- Typically uses **4 bytes (32 bits)**.

- Format: 1 bit for sign, 8 bits for exponent, 23 bits for mantissa (significand).
- Precision: ~6–7 decimal digits.
- Represents numbers in scientific notation (e.g., 1.23×1041.23 \times 10^41.23×104).

Use Case Example:

c
CopyEdit

```
float voltage = 3.3;
```

Binary Representation (IEEE 754):

- `float f = 3.14;` → `01000000 01001000 11110111 11011111`

🔹 `char` – Character

Definition:
Stores a **single character**, often encoded using **ASCII** or **UTF-8**.

Key Characteristics:

- Occupies **1 byte (8 bits)**.
- Can also be interpreted as an 8-bit integer.
- Range: 0 to 255 (unsigned), or -128 to 127 (signed), depending on compiler.

Use Case Example:

c
CopyEdit

```
char grade = 'A';
```

Binary Representation:

- `'A'` in ASCII → Decimal: `65` → Binary: `01000001`

2 . What is the role of #include in a C program?

The line `#include <stdio.h>` in a C program tells the compiler to include the **Standard Input Output** header file before compiling the program.

Purpose of `#include <stdio.h>`:

- It provides the declarations for standard input and output functions such as:
  - `printf()` – for outputting text to the console
  - `scanf()` – for reading input from the user
  - `getchar()`, `putchar()`, `fopen()`, `fclose()`, `fread()`, `fwrite()`, etc.

Without including `stdio.h`, if you try to use these functions, the compiler will not recognize them, leading to errors or warnings.

3. List and explain any 3 rules for naming variables in C

---

1. Variable names must begin with a letter (A–Z or a–z) or an underscore (_)

- ✅ Valid: `count`, `_temp`, `Value1`
- ❌ Invalid: `1value`, `@num`
- **Explanation**: A variable cannot start with a digit or special characters (except underscore).

---

2. Variable names can only contain letters, digits (0–9), and underscores

- ✅ Valid: `total_sum`, `value123`, `x_1`
- ❌ Invalid: `total-sum`, `value!`, `x&y`
- **Explanation**: Hyphens, spaces, and other special characters are not allowed.

---

3. Variable names cannot be the same as C keywords

- ❌ Invalid: `int`, `return`, `for` (These are reserved by the language)
- ✅ Valid: `count`, `index`, `loop`
- **Explanation**: Keywords have predefined meanings in C and cannot be used as variable names.

4.What is the use of the return 0; statement in the main() function?

The statement `return 0;` in the `main()` function of a C program is used to:

Indicate successful program termination.

Here's what it does:

- When `main()` finishes running, it **returns a value to the operating system**.
- `return 0;` means **the program ended without errors**.
- A **non-zero return value** (e.g., `return 1;`) typically signals an **error or abnormal termination**.

Example:

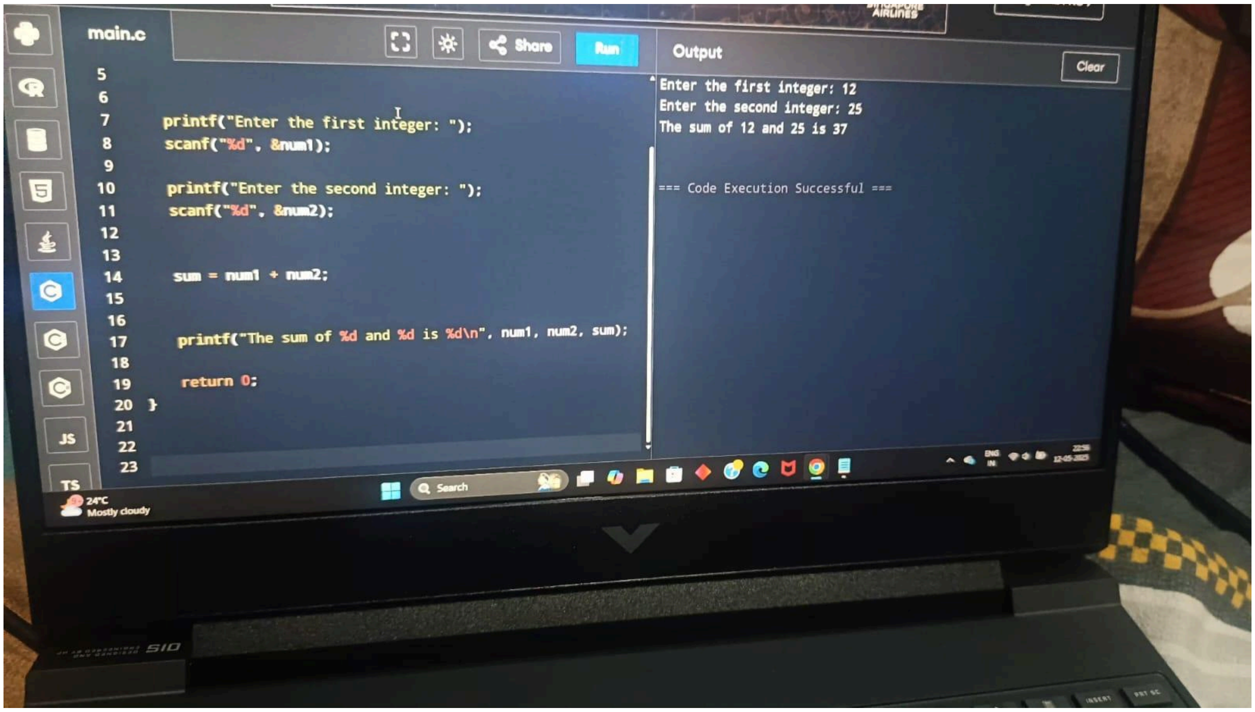
c

CopyEdit

```
#include <stdio.h>
```

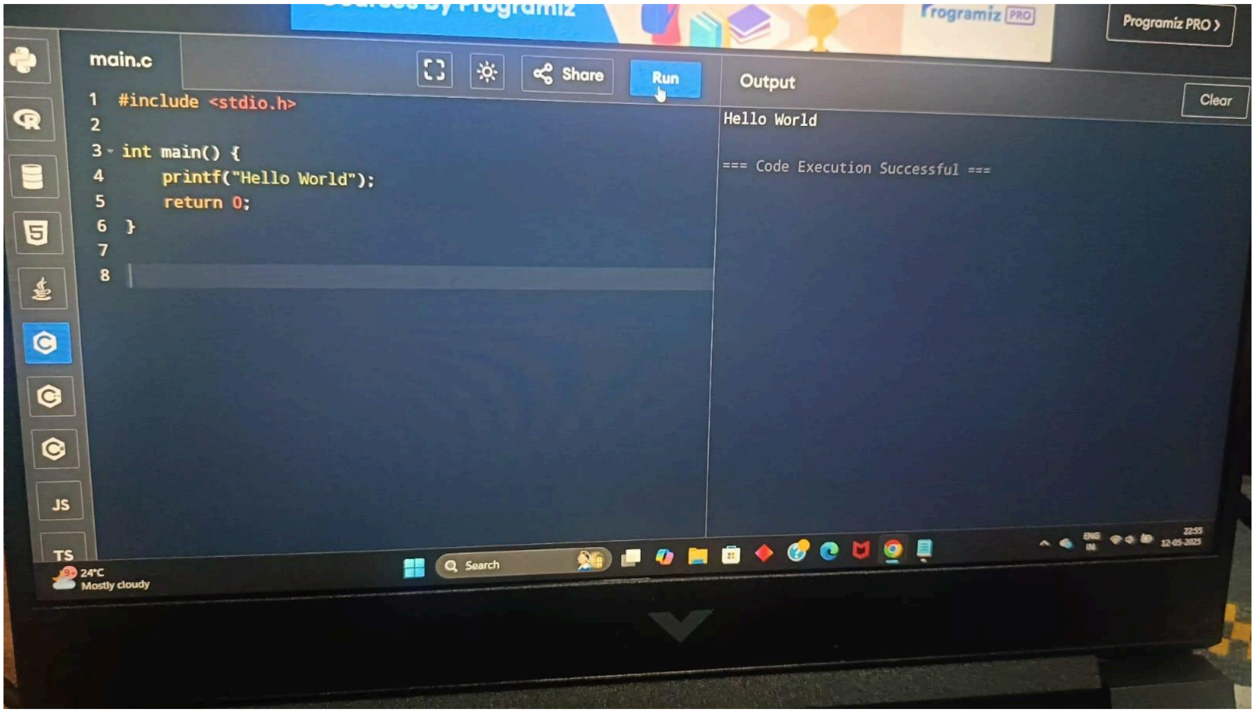
```
int main() {
    printf("Hello, world!\n");
    return 0;
}
```

In this example, `return 0;` tells the system that everything ran correctly.



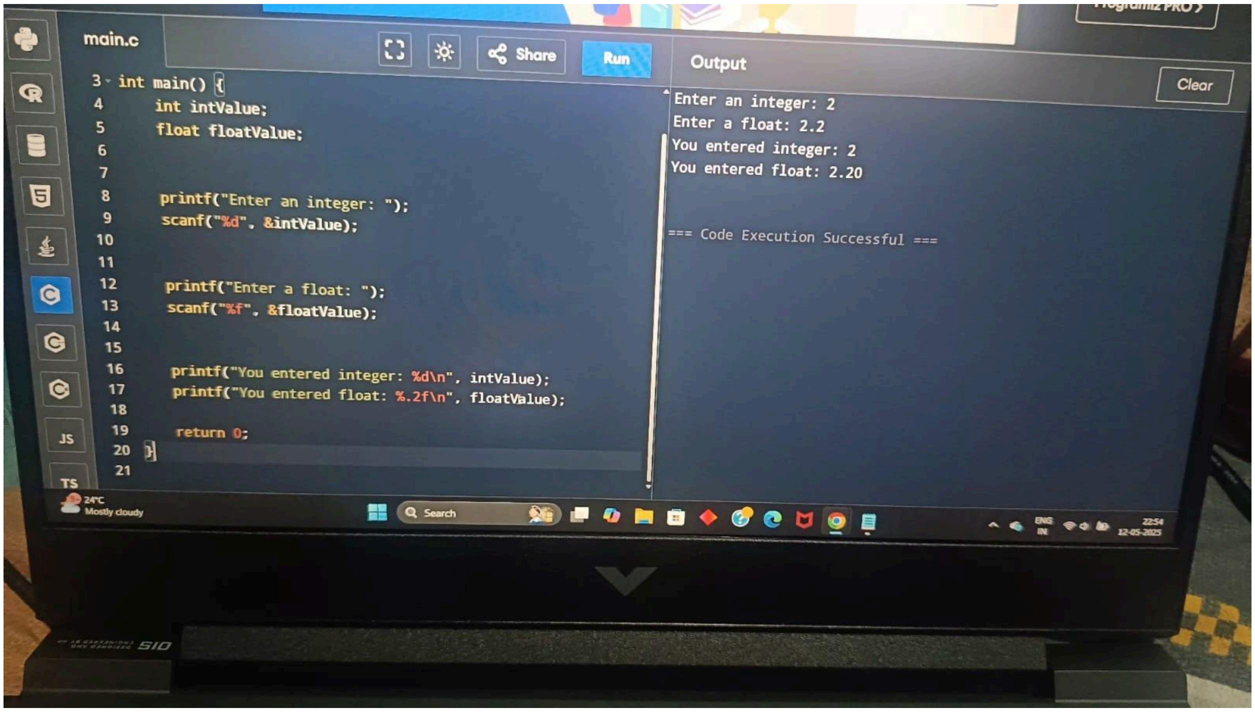
```c
    printf("Enter the first integer: ");
    scanf("%d", &num1);

    printf("Enter the second integer: ");
    scanf("%d", &num2);


    sum = num1 + num2;


    printf("The sum of %d and %d is %d\n", num1, num2, sum);

    return 0;
}
```

```
Enter the first integer: 12
Enter the second integer: 25
The sum of 12 and 25 is 37


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



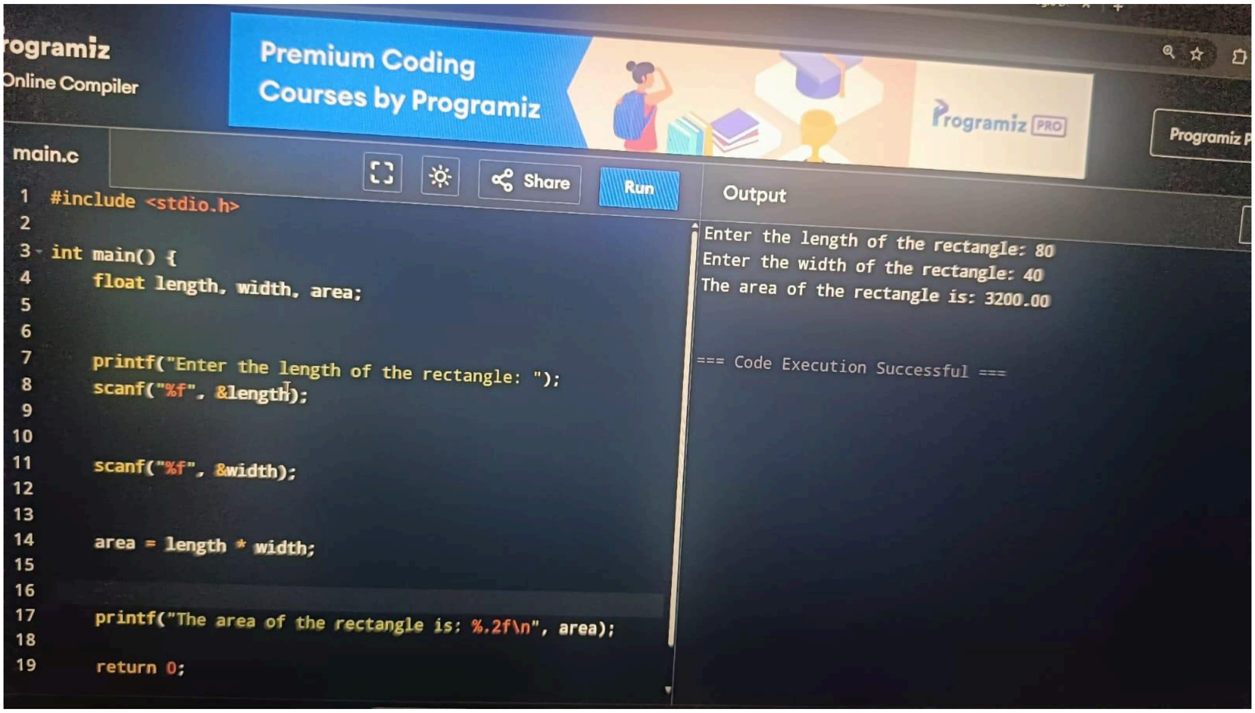
```c
#include <stdio.h>

int main() {
    printf("Hello World");
    return 0;
}
```

```
Hello World

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



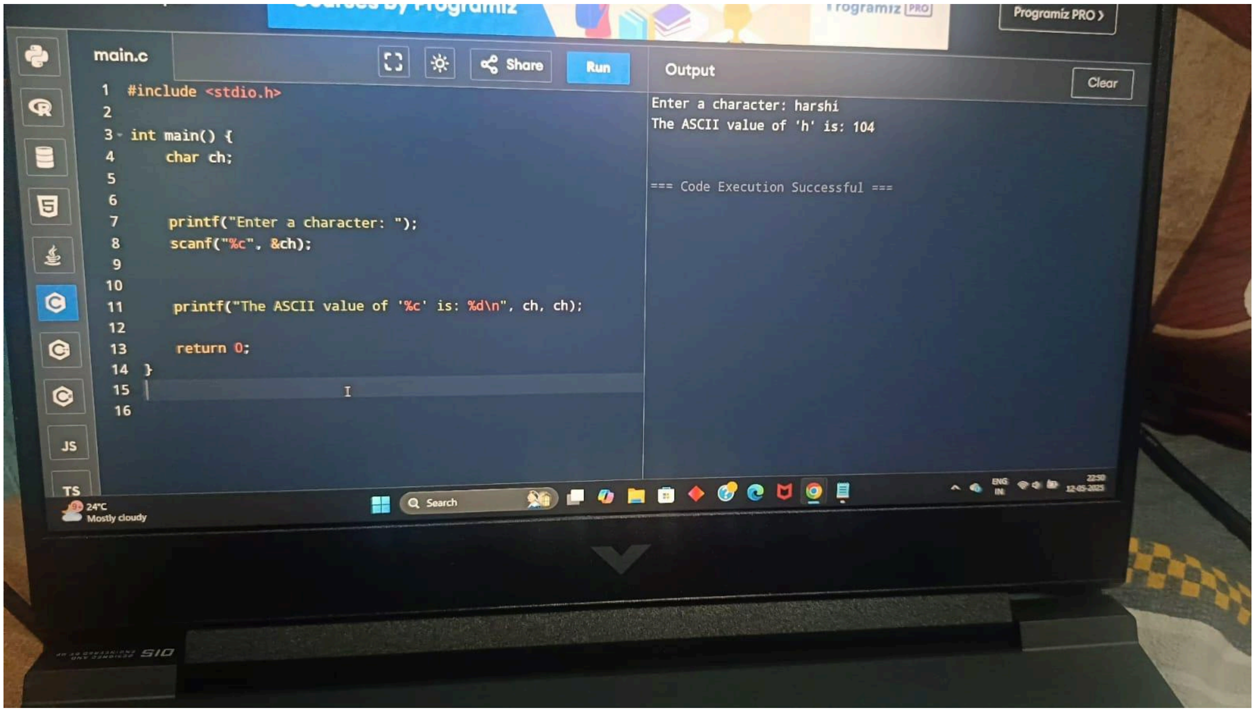
```c
int main() {
    int intValue;
    float floatValue;


    printf("Enter an integer: ");
    scanf("%d", &intValue);


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


    printf("You entered integer: %d\n", intValue);
    printf("You entered float: %.2f\n", floatValue);

    return 0;
}
```

```
Enter an integer: 2
Enter a float: 2.2
You entered integer: 2
You entered float: 2.20


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



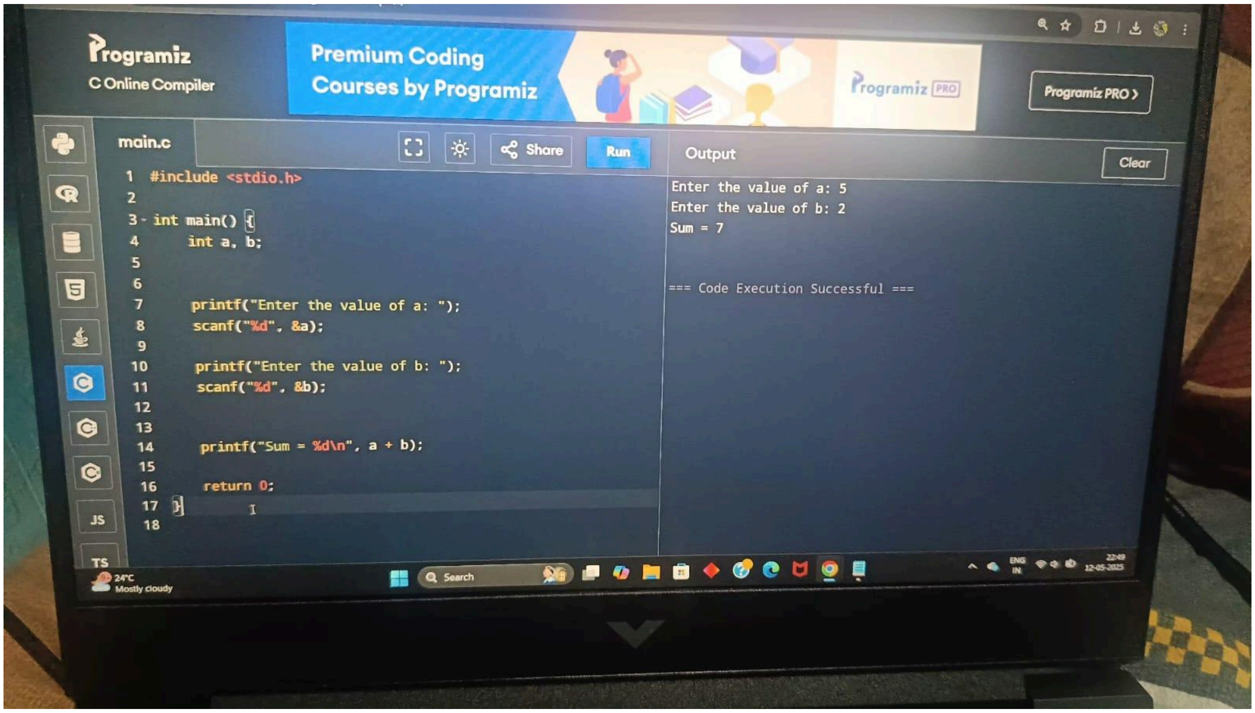
```c
#include <stdio.h>

int main() {
    float length, width, area;


    printf("Enter the length of the rectangle: ");
    scanf("%f", &length);


    scanf("%f", &width);


    area = length * width;


    printf("The area of the rectangle is: %.2f\n", area);

    return 0;
```

```
Enter the length of the rectangle: 80
Enter the width of the rectangle: 40
The area of the rectangle is: 3200.00


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



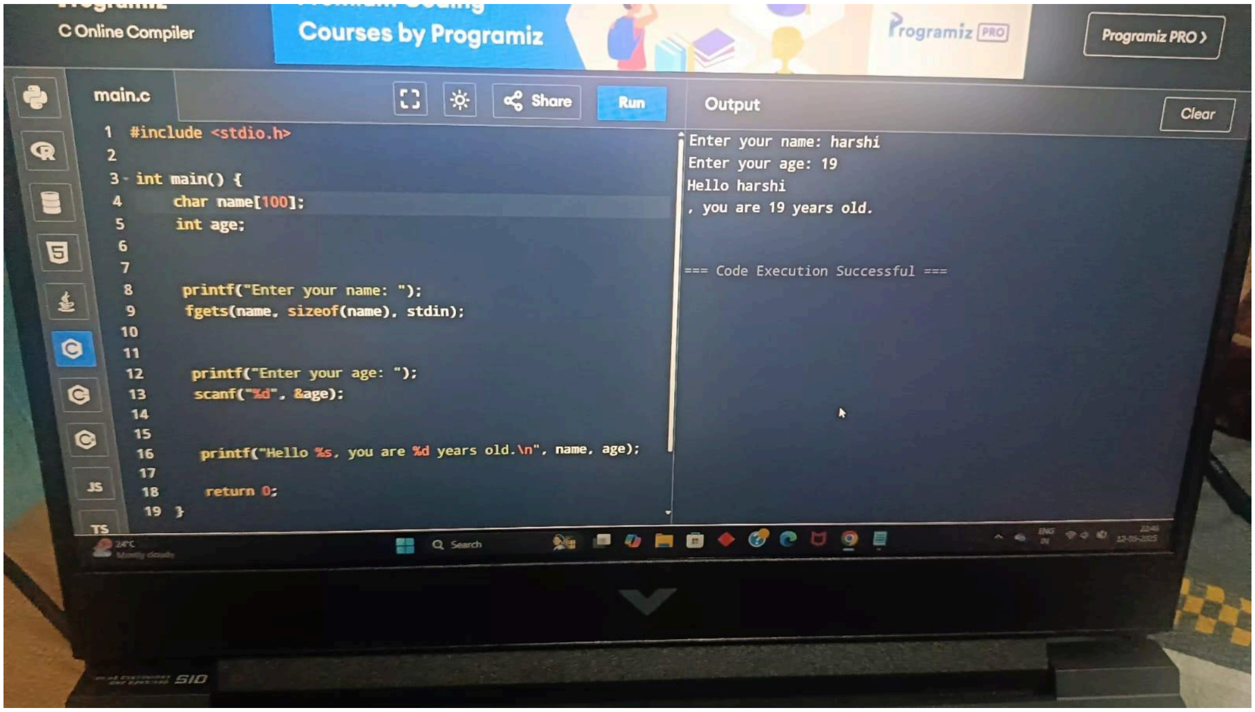
```c
#include <stdio.h>

int main() {
    char ch;


    printf("Enter a character: ");
    scanf("%c", &ch);


    printf("The ASCII value of '%c' is: %d\n", ch, ch);

    return 0;
}

```

```
Enter a character: harshi
The ASCII value of 'h' is: 104


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



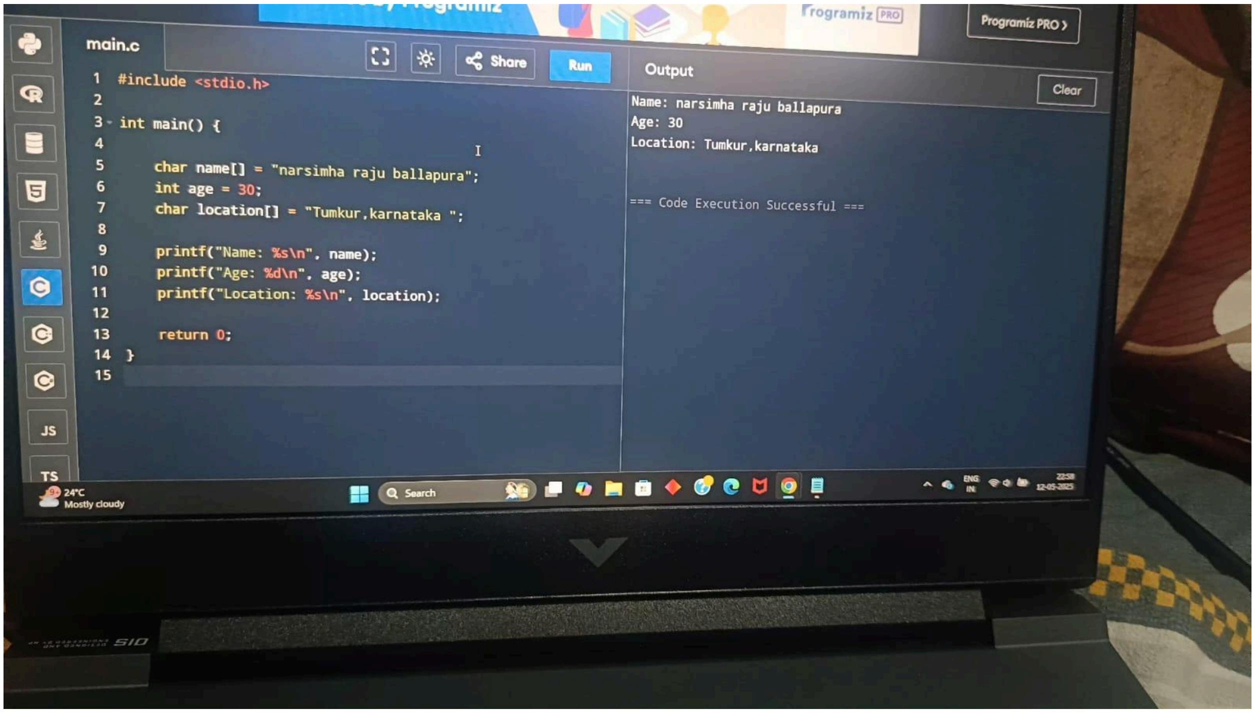
```c
#include <stdio.h>

int main() {
    int a, b;


    printf("Enter the value of a: ");
    scanf("%d", &a);

    printf("Enter the value of b: ");
    scanf("%d", &b);


    printf("Sum = %d\n", a + b);

    return 0;
}

```

```
Enter the value of a: 5
Enter the value of b: 2
Sum = 7


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



```c
#include <stdio.h>

int main() {
    char name[100];
    int age;


    printf("Enter your name: ");
    fgets(name, sizeof(name), stdin);


    printf("Enter your age: ");
    scanf("%d", &age);


    printf("Hello %s, you are %d years old.\n", name, age);

    return 0;
}
```

```
Enter your name: harshi
Enter your age: 19
Hello harshi
, you are 19 years old.


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



```c
#include <stdio.h>

int main() {

    char name[] = "narsimha raju ballapura";
    int age = 30;
    char location[] = "Tumkur,karnataka ";

    printf("Name: %s\n", name);
    printf("Age: %d\n", age);
    printf("Location: %s\n", location);

    return 0;
}
```

```
Name: narsimha raju ballapura
Age: 30
Location: Tumkur,karnataka


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