Practice Question

- By Aditya Varma

MCQs

1. What is the primary function of the "main" function in a C program?
   a. Execution starts from here
   b. Declaration of variables
   c. Printing output
   d. Handling errors
   Answer: a

2. Which symbol is used to indicate a preprocessor directive in C?
   a. @
   b. #
   c. $
   d. &
   Answer: b

3. What is the size of the "int" data type in C?
   a. 2 bytes
   b. 4 bytes
   c. 8 bytes
   d. Depends on the system
   Answer: b

4. In C, how do you declare a constant?
   a. const var;
   b. #define var;
   c. final var;
   d. constant var;
   Answer: a

5. What is the purpose of the "sizeof" operator in C?
   a. Returns the size of a variable or data type
   b. Finds the square root of a number
   c. Concatenates two strings
   d. Converts a variable to a string
   Answer: a

6. What is the correct syntax for a single-line comment in C?
a. // Comment
b. /* Comment */
c. # Comment #
d. -- Comment --
Answer: a

7. How do you include a header file named "stdio.h" in a C program?
a. #include <stdio.h>
b. import stdio.h;
c. <stdio.h>
d. include <stdio.h>
Answer: a

8. What is the purpose of the "scanf" function in C?
a. Print formatted output
b. Read input from the user
c. Calculate square root
d. Allocate memory
Answer: b

9. In C, what is the relationship between arrays and pointers?
a. They are unrelated
b. Arrays are pointers
c. Pointers are arrays
d. Both a and b
Answer: d

10. How do you dynamically allocate memory in C?
a. malloc()
b. allocate()
c. new()
d. create()
Answer: a

11. What is the purpose of the "break" statement in C?
a. Exit the loop
b. Skip the current iteration
c. Jump to a specified label
d. Terminate the program
Answer: a

12. Which function is used to open a file in C?
    a. open()
    b. fopen()
    c. file_open()
    d. readfile()
    Answer: b

13. What is the correct way to compare two strings in C?
    a. strcomp()
    b. strcmp()
    c. compare()
    d. string_compare()
    Answer: b

14. How do you define a macro in C?
    a. macro myMacro {}
    b. define myMacro {}
    c. #define myMacro {}
    d. macro = myMacro
    Answer: c

15. What is the purpose of the "typedef" keyword in C?
    a. Define a new data type
    b. Declare a variable
    c. Create an alias for a data type
    d. Include a header file
    Answer: a

16. In C, what is the purpose of the "continue" statement?
    a. End the loop
    b. Skip the remaining code in the loop and continue with the next iteration
    c. Jump to a specified label
    d. Terminate the program
    Answer: b

17. What is the result of the expression: 5 + 3 * 2?
    a. 16
    b. 11
    c. 13
    d. 26
    Answer: a

18. Which escape sequence is used to represent a newline character in C?
   a. \n
   b. \r
   c. \t
   d. \b
   Answer: a

19. What is the correct way to declare a function in C?
   a. function myFunction() {}
   b. declare myFunction() {}
   c. void myFunction() {}
   d. define myFunction() {}
   Answer: c

20. How do you pass an array to a function in C?
   a. Pass the entire array
   b. Pass the array size
   c. Pass the first element of the array
   d. Pass a pointer to the array
   Answer: d

21. What does the "static" keyword do in C?
   a. Allocate memory dynamically
   b. Make a variable local to the file
   c. Define a constant
   d. Declare a function
   Answer: b

22. What is the purpose of the "fclose" function in C?
   a. Close a file
   b. Open a file
   c. Read from a file
   d. Write to a file
   Answer: a

23. Which bitwise operator is used for the right shift in C?
   a. <<
   b. >>
   c. &
   d. |
   Answer: b

24. How do you declare a two-dimensional array in C?
    a. int arr[][];
    b. int arr;
    c. int arr[];
    d. int arr[][]
    Answer: a

25. What is the purpose of the "union" in C?
    a. Combine two structures
    b. Define a new data type
    c. Declare a variable
    d. Create an alias for a data type
    Answer: a

26. Which function is used to print formatted output in C?
    a. printf()
    b. print()
    c. format()
    d. display()
    Answer: a

27. How do you initialize a character array with the string "Hello" in C?
    a. char str[5] = "Hello";
    b. char str[] = "Hello";
    c. char str[6] = "Hello";
    d. char str[5] = {'H', 'e', 'l', 'l', 'o'};
    Answer: b

28. What is the purpose of the "do-while" loop in C?
    a. Iterate until a condition is true
    b. Iterate at least once, then check the condition
    c. Iterate based on a counter
    d. Iterate indefinitely
    Answer: b

29. How do you access the value of a variable through a pointer in C?
    a. *var
    b. var*
    c. &var
    d. var&
    Answer: a

30. Which function is used to convert a string to an integer in C?

a. atoi()
b. itoa()
c. str2int()
d. int2str()
Answer: a

31. What is the purpose of the "volatile" keyword in C?

a. Make a variable constant
b. Declare a function
c. Indicate that a variable may be changed by external factors
d. Define a constant
Answer: c

32. How do you allocate memory for an array dynamically in C?

a. allocate()
b. new()
c. malloc()
d. create()
Answer: c

33. What is the correct way to declare a pointer in C?

a. int ptr;
b. ptr int;
c. int* ptr;
d. pointer int;
Answer: c

34. What is the purpose of the "const" keyword in C?

a. Define a constant
b. Declare a variable
c. Make a variable constant
d. Include a header file
Answer: c

35. Which operator is used to access the value pointed to by a pointer in C?

a. *
b. &
c. ->
d. ::
Answer: a

36. How do you declare a structure in C?
    a. struct myStruct {};
    b. structure myStruct {};
    c. define myStruct {};
    d. create myStruct {};
    Answer: a

37. What is the purpose of the "goto" statement in C?
    a. Jump to a specified label
    b. Terminate the program
    c. Exit the loop
    d. Skip the remaining code in the loop and continue with the next iteration
    Answer: a

38. Which library function is used to find the length of a string in C?
    a. length()
    b. size()
    c. strlen()
    d. strlength()
    Answer: c

39. What is the correct syntax for a multi-line comment in C?
    a. /* Comment */
    b. // Comment //
    c. # Comment #
    d. -- Comment --
    Answer: a

40. How do you define a constant pointer in C?
    a. const int *ptr;
    b. int const *ptr;
    c. int *const ptr;
    d. const int* const ptr;
    Answer: d

41. What is the purpose of the "rand" function in C?
    a. Find the remainder of a division
    b. Generate a random number
    c. Return the absolute value of a number
    d. Round a floating-point number
    Answer: b

42. How do you declare a variable that can hold the address of any data type in C?

a. generic var;
b. anytype var;
c. void* var;
d. dynamic var;
Answer: c

43. What is the purpose of the "memset" function in C?

a. Copy memory from one location to another
b. Set a block of memory to a specific value
c. Allocate memory dynamically
d. Free allocated memory
Answer: b

44. In C, what is the purpose of the "sizeof" operator when used with a structure?

a. Return the size of the structure
b. Find the number of elements in the structure
c. Calculate the sum of elements in the structure
d. Find the length of a string in the structure
Answer: a

45. What is the purpose of the "const" qualifier in a function declaration in C?

a. Indicate that the function is constant
b. Declare a constant function
c. Specify that the function does not modify its parameters
d. Define a constant variable
Answer: c

46. How do you declare a multi-dimensional array in C?

a. int arr[][[;
b. int arr[][][];
c. int arr[]];
d. int arr[][][]
Answer: b

47. What is the purpose of the "feof" function in C?

a. Test the end-of-file indicator for a file
b. Open a file
c. Close a file
d. Read from a file
Answer: a

48. How do you include the contents of one file in another file in C?

   a. #include <filename>

   b. import filename;

   c. include <filename>

   d. load filename;

   Answer: a

49. What is the purpose of the "strcat" function in C?

   a. Compare two strings

   b. Copy one string to another

   c. Concatenate two strings

   d. Find the length of a string

   Answer: c

50. How do you return a value from a function in C?

   a. return value;

   b. value return;

   c. exit(value);

   d. value exit;

   Answer: a

51. Which of the following is a valid variable name in C?

   a. 2value

   b. value_2

   c. float

   d. &total

   Answer: b

52. What will printf("%d", 10/4); output?

   a. 2.5

   b. 2

   c. 2.0

   d. 2.25

   Answer: b

53. Which keyword is used to return a value from a function in C?

   a. return

   b. break

   c. goto

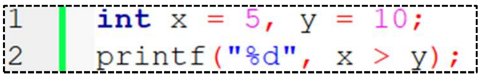
   d. end

   Answer: a

54. Which header file is needed for printf and scanf functions?
    a. conio.h
    b. stdlib.h
    c. string.h
    d. stdio.h
    Answer: d

55. In C, what is the index of the first element in an array?
    a. -1
    b. 0
    c. 1
    d. Depends on array type
    Answer: b

56. What will the following code output?



```
int x = 5, y = 10;
printf("%d", x > y);
```

    a. 5
    b. 10
    c. 1
    d. 0
    Answer: d

57. Which of the following loops will execute at least once?
    a. for loop
    b. while loop
    c. do-while loop
    d. None
    Answer: c

58. How can you pass a value to a function in C?
    a. Call by function
    b. Call by value
    c. Call by keyword
    d. Call by structure
    Answer: b

59. Which operator is used to find remainder in C?

a. /
b. %
c. *
d. //
Answer: b

60. What is the output of printf("%c", 'A' + 2);?

a. A
b. B
c. C
d. D
Answer: c

61. Which function is used to find the length of a string in C?

a. strlength()
b. strcount()
c. strlen()
d. countstr()
Answer: c

62. What does this condition mean: if(a == b) ?

a. Assign b to a
b. Compare a and b
c. Add a and b
d. Subtract b from a
Answer: b

63. Which of the following data types can store a single character?

a. int
b. char
c. float
d. double
Answer: b

64. Which loop is best when the number of iterations is known?

a. while
b. do-while
c. for
d. if
Answer: c

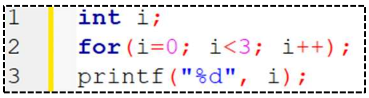
65. Which function is used to take a single character input in C?

a. getchar()
b. gets()
c. scanf()
d. putchar()
Answer: a

66. Which of these correctly declares a 1D array of 5 integers?

a. int arr(5);
b. int arr[5];
c. int arr;
d. int arr{5};
Answer: b

67. What is the output of the code?



```
int i;
for(i=0; i<3; i++);
printf("%d", i);
```

a. 0
b. 2
c. 3
d. Error
Answer: c

68. What will the expression 7 % 3 evaluate to?

a. 1
b. 2
c. 3
d. 4
Answer: a

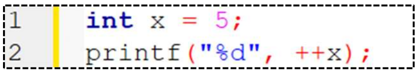
69. In C, a function that does not return any value must be declared as:

a. void function()
b. empty function()
c. null function()
d. none
Answer: a

70. Which of the following correctly declares a 2D array?

a. int arr[3,4];
b. int arr(3,4);
c. int arr[3][4];
d. int arr[3-4];
Answer: c

71. What will be the output?



```
int x = 5;
printf("%d", ++x);
```

a. 4
b. 5
c. 6
d. Error
Answer: c

72. Which statement is used to exit from a loop immediately?

a. skip
b. stop
c. break
d. continue
Answer: c

73. Which keyword is used to prevent a variable from being modified?

a. fixed
b. static
c. const
d. final
Answer: c

74. What will printf("%d", 2 && 0); print?

a. 0
b. 1
c. 2
d. Error
Answer: a

75. How do you define a user-defined function in C?
    a. By using the keyword function
    b. By writing a separate block with return type
    c. By using macro
    d. You can't define one
    Answer: b

76. What will printf("%d", 3==3); print?
    a. 3
    b. 1
    c. 0
    d. Error
    Answer: b

77. In which memory section are arrays stored in C?
    a. Register
    b. Heap
    c. Stack
    d. ROM
    Answer: c

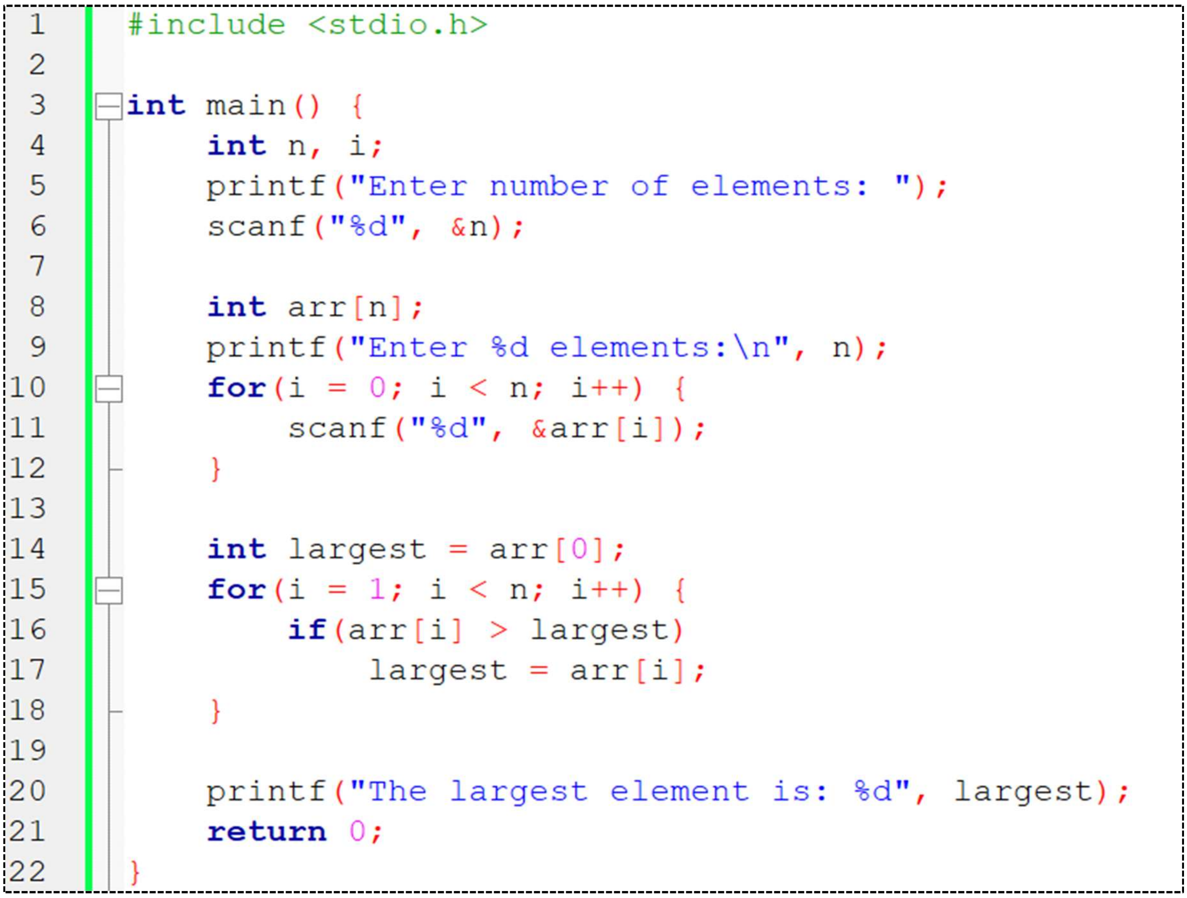
78. Which function call is correct if a function is defined as int sum(int a, int b);?
    a. sum();
    b. sum(a, b);
    c. sum(3, 4);
    d. Both b and c
    Answer: d

79. What does continue do inside a loop?
    a. Stops loop completely
    b. Skips current iteration
    c. Ends the program
    d. Returns a value
    Answer: b

80. What will happen if you don't write a return type for a function in C?
    a. It will default to int
    b. It will cause a compile-time error
    c. It will default to void
    d. It will not compile
    Answer: a

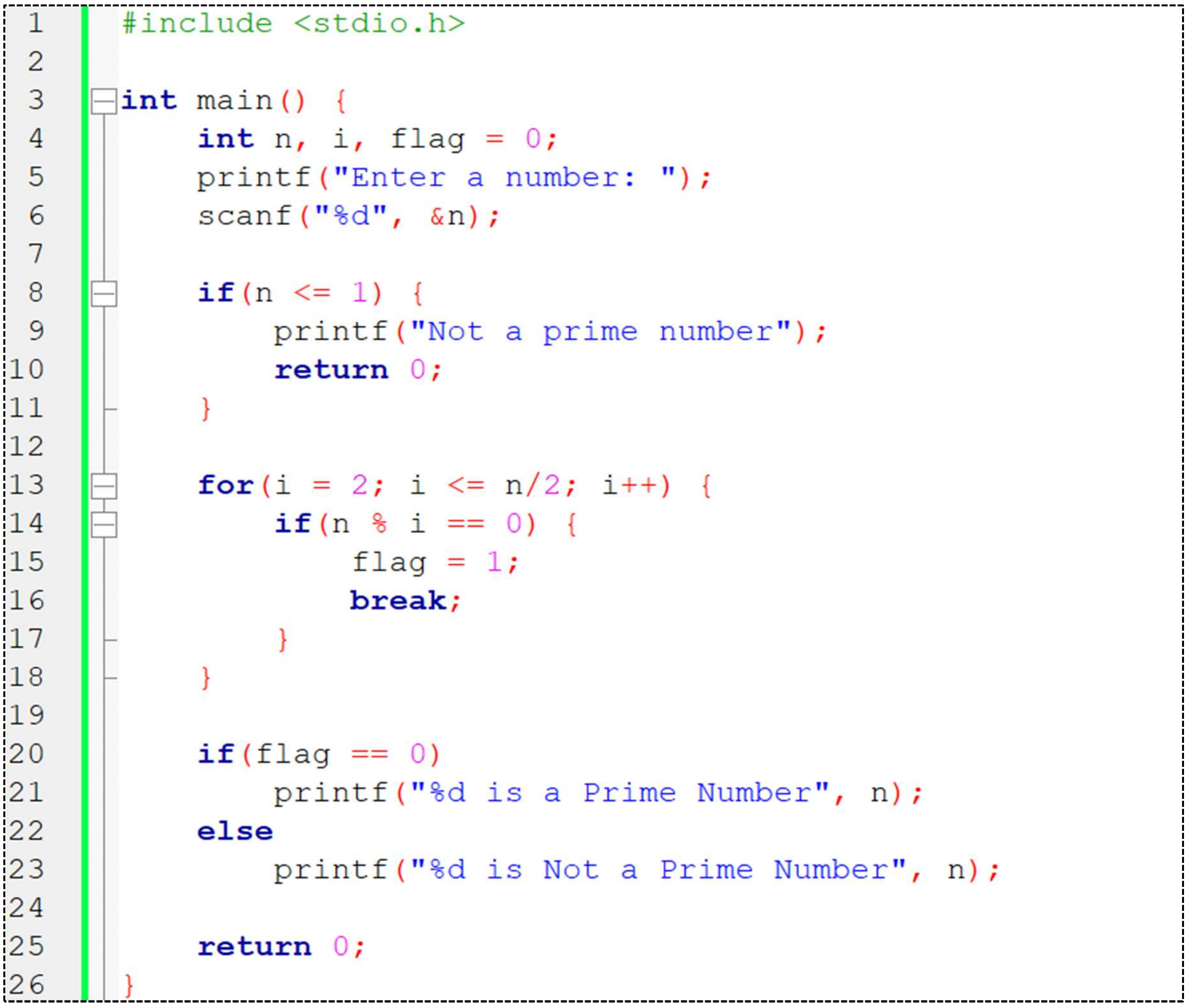
Programs

Program 1: Program to Find the Largest Element in an Array
Solution:



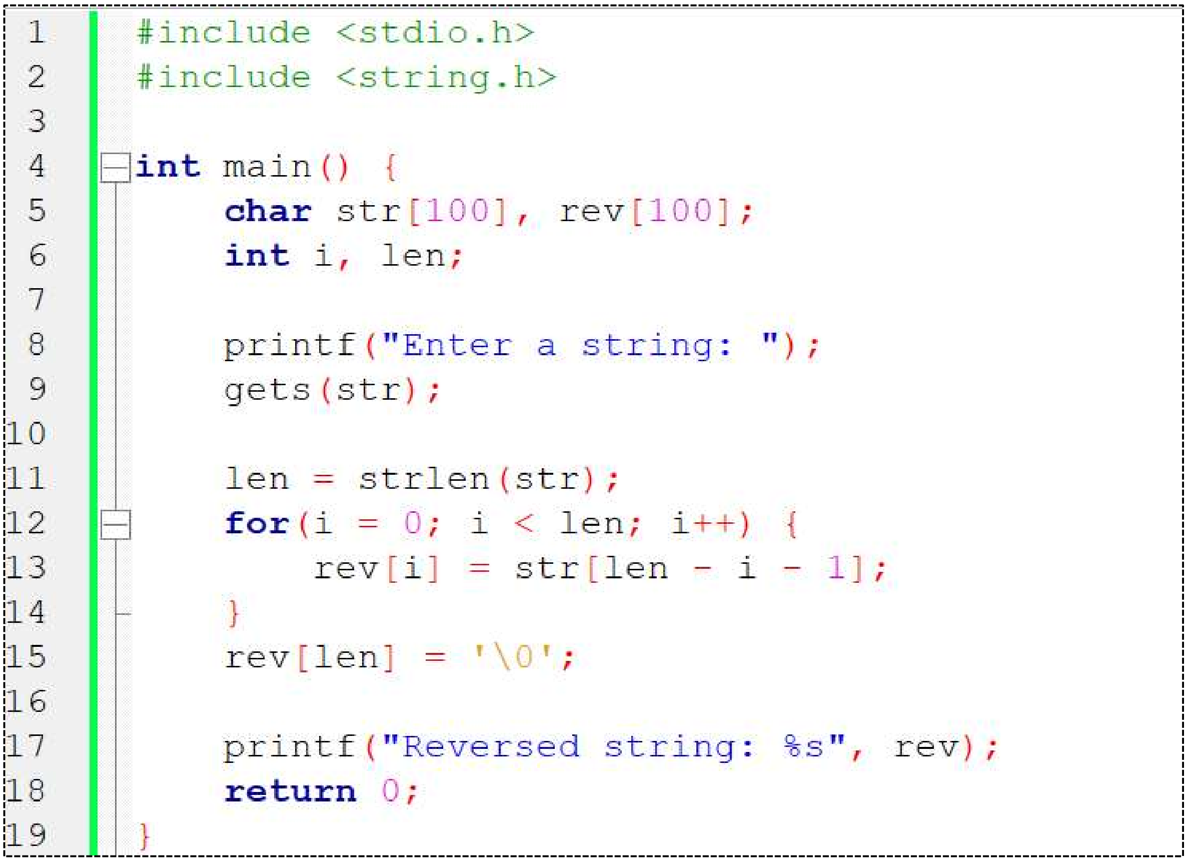
```c
#include <stdio.h>

int main() {
    int n, i;
    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int largest = arr[0];
    for(i = 1; i < n; i++) {
        if(arr[i] > largest)
            largest = arr[i];
    }

    printf("The largest element is: %d", largest);
    return 0;
}
```

Program 2: Program to Check Whether a Number is Prime
Solution:



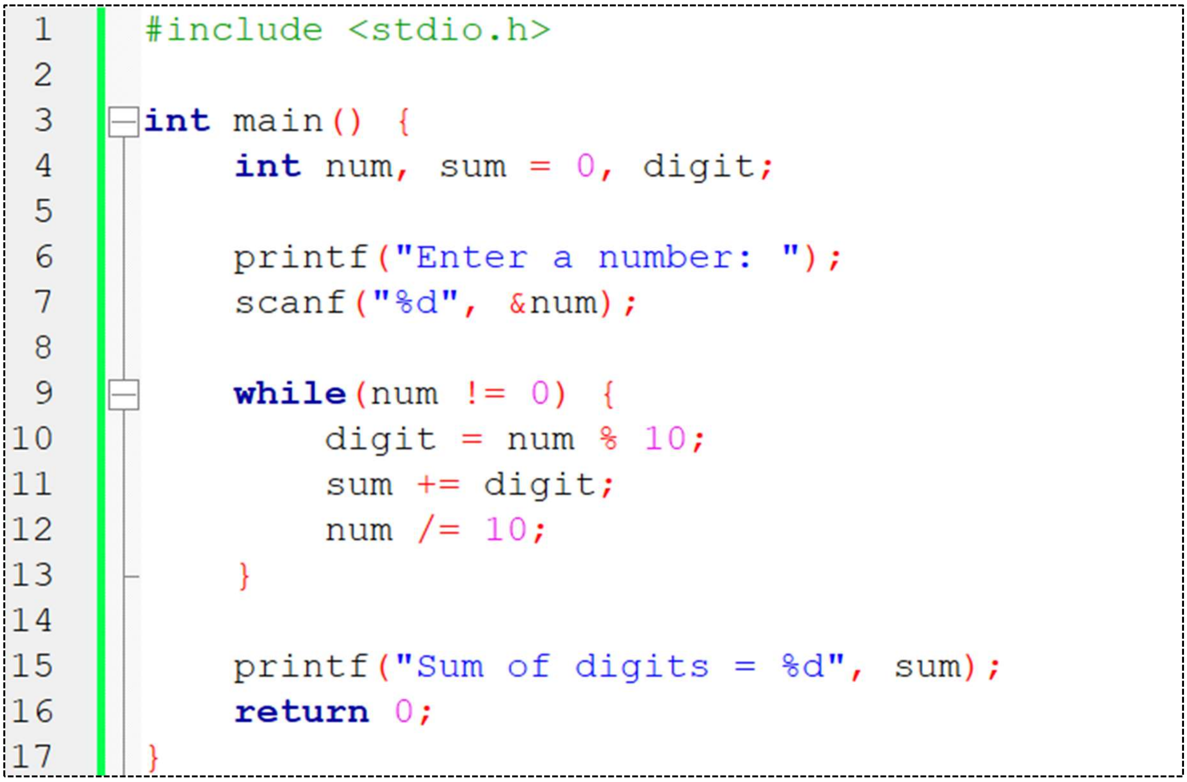
```c
#include <stdio.h>

int main() {
    int n, i, flag = 0;
    printf("Enter a number: ");
    scanf("%d", &n);

    if(n <= 1) {
        printf("Not a prime number");
        return 0;
    }

    for(i = 2; i <= n/2; i++) {
        if(n % i == 0) {
            flag = 1;
            break;
        }
    }

    if(flag == 0)
        printf("%d is a Prime Number", n);
    else
        printf("%d is Not a Prime Number", n);

    return 0;
}
```

Program 3: Program to Reverse a String
Solution:



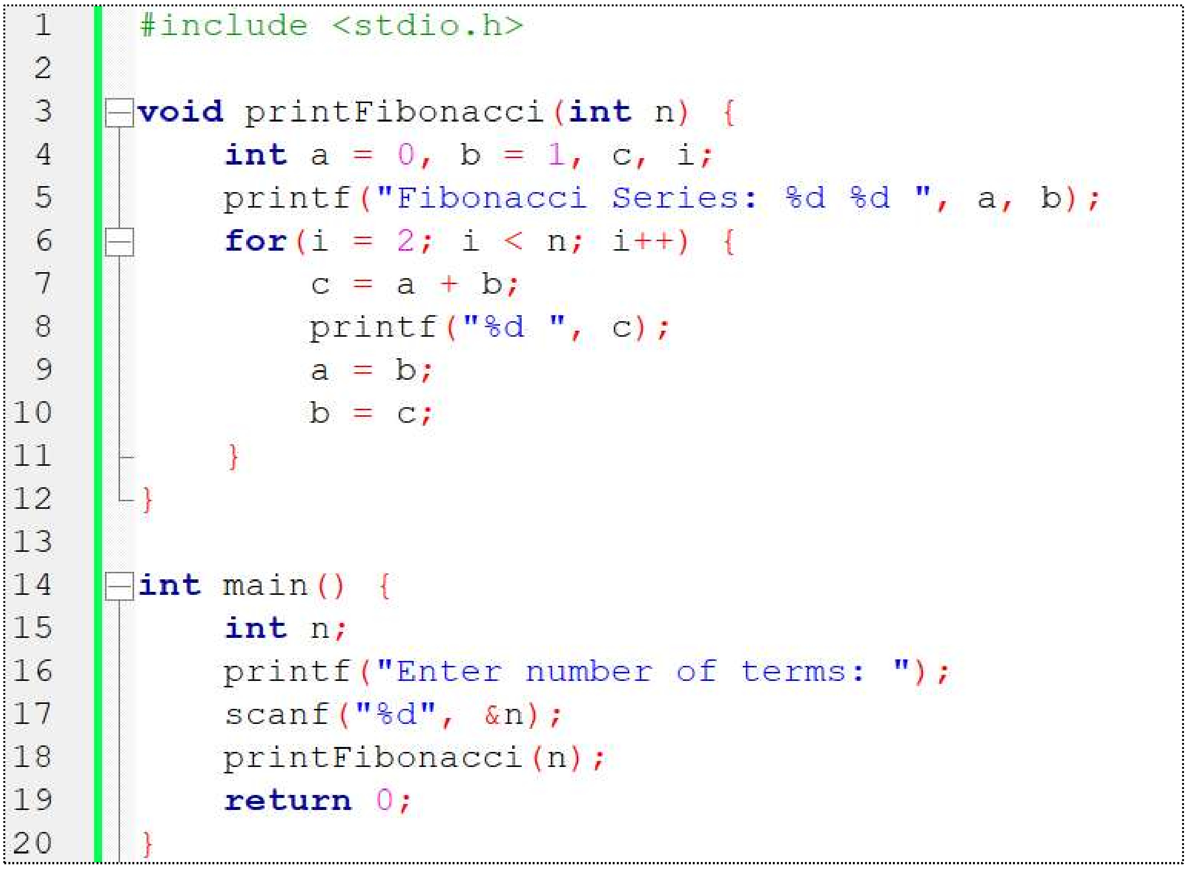
```c
#include <stdio.h>
#include <string.h>

int main() {
    char str[100], rev[100];
    int i, len;

    printf("Enter a string: ");
    gets(str);

    len = strlen(str);
    for(i = 0; i < len; i++) {
        rev[i] = str[len - i - 1];
    }
    rev[len] = '\0';

    printf("Reversed string: %s", rev);
    return 0;
}
```

Program 4: Program to Find Sum of Digits of a Number
Solution:



```c
#include <stdio.h>

int main() {
    int num, sum = 0, digit;

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

    while(num != 0) {
        digit = num % 10;
        sum += digit;
        num /= 10;
    }

    printf("Sum of digits = %d", sum);
    return 0;
}
```

Program 5: Program to Find Fibonacci Series Using Function
Solution:



```c
#include <stdio.h>

void printFibonacci(int n) {
    int a = 0, b = 1, c, i;
    printf("Fibonacci Series: %d %d ", a, b);
    for(i = 2; i < n; i++) {
        c = a + b;
        printf("%d ", c);
        a = b;
        b = c;
    }
}

int main() {
    int n;
    printf("Enter number of terms: ");
    scanf("%d", &n);
    printFibonacci(n);
    return 0;
}
```