Comprehensive C Programming MCQ Question Paper

Table of Contents

Introduction to C Programming

1. Who developed the C programming language?

a) Bill Gates
b) Dennis Ritchie
c) Steve Jobs
d) Linus Torvalds

2. In which year was the C programming language developed?

a) 1969-1973
b) 1975-1980

c) 1980-1985
d) 1985-1990

3. What is the primary purpose of the main() function in a C program?

a) To declare variables
b) To define constants
c) To serve as the entry point of the program
d) To include header files

4. Which of the following is NOT a key feature of C programming language?

a) Fast performance
b) Direct memory access
c) Automatic garbage collection
d) Works everywhere

5. What does IDE stand for in the context of programming?

a) Integrated Development Environment
b) Intelligent Development Engine
c) Interactive Design Environment
d) Integrated Debugging Environment

6. Which of the following is recommended as a beginner-friendly IDE for C programming?

a) Visual Studio Code
b) C-Free
c) Eclipse
d) NetBeans

7. What is the file extension for a C source code file?

a) .cpp
b) .cs

c) .c
d) .class

8. What does the return 0; statement in the main() function indicate?

a) The program has an error
b) Successful program termination
c) The program needs to restart
d) Memory allocation failure

Variables and Data Types

9. What is a variable in C programming?

a) A constant value
b) A storage location with a name that holds data
c) A function name
d) A loop counter

10. Which data type is used to store a single character in C?

```
a) int
b) float
c) char
d) double
```

11. What is the format specifier for int data type in printf() function?

```
a) %f
b) %c
c) %d
```

```
d) %s
```

12. What is the size of an int data type in bytes (typically)?

```
a) 1
b) 2
c) 4
d) 8
```

13. Which data type is used to store decimal numbers in C?

```
a) int
b) char
c) float
d) string
```

14. What is the format specifier for float data type in printf() function?

```
a) %d
b) %f
c) %c
d) %s
```

15. What does the format specifier %.2f do?

```
a) Displays 2 decimal places
b) Displays 2 digits before decimal point
c) Multiplies the number by 2
d) Rounds the number to nearest 2
```

16. Which of the following is the correct way to declare and initialize a character variable with the value 'A'?

```
a) char ch = "A";
b) char ch = 'A';
c) char ch = A;
d) char ch = 'A';
```

17. What is the range of char data type?

```
a) 0 to 255
b) -128 to 127
c) -32768 to 32767
d) 0 to 65535
```

18. Which data type would you use to store the value 3.14159265359 with maximum precision?

```
a) float
b) int
c) double
d) char
```

19. What is the size of double data type in bytes (typically)?

```
a) 4
b) 8
c) 16
d) 2
```

20. What does the format specifier %lf represent in printf()?

```
a) %d
b) %f
c) %lf
d) %df
```

Operators

21. Which operator is used for addition in C?

```
a) +
b) -
c) *
d) /
```

22. Which operator is used for multiplication in C?

```
a) x
b) *
c) ×
d) ·
```

23. Which operator is used for division in C?

```
a) ÷
b) /
c) :
d) %
```

24. What does the % operator do in C?

```
a) Percentage calculation
b) Modulus (remainder)
c) Multiplication
d) Division
```

25. What is the result of 7 % 3?

```
a) 2
b) 1
c) 2.33
d) 3
```

26. What is the result of 10 / 3 in C when both operands are integers?

```
a) 3.33
b) 3
c) 4
d) 1
```

27. Which operator is used for comparison "equal to" in C?

```
a) =
b) ==
c) !=
d) ===
```

28. Which operator is used for "not equal to" in C?

```
a) <>
b) !=
c) =!
d) !==
```

29. What is the result of 5 > 3 in C?

```
a) 1
b) 0
c) True
d) False
```

30. In C, what does 0 represent in boolean context?

```
a) True
b) False
c) Undefined
d) Error
```

31. What is the result of (5 > 3) && (2 < 4)?

```
a) 1
b) 0
c) True
d) False
```

32. What is the result of (5 > 10) || (2 < 4)?

```
a) 1
b) 0
c) True
d) False
```

Input/Output Operations

33. Which function is used to display output to the screen in C?

```
a) scanf()
b) printf()
c) display()
d) output()
```

34. Which function is used to read input from the keyboard in C?

```
a) printf()
b) scanf()
c) input()
d) read()
```

35. What is the purpose of the & operator in scanf()?

```
a) To multiply numbers
b) To get the address of a variable
c) To compare values
d) To concatenate strings
```

36. Which format specifier is used to read an integer using scanf()?

```
a) %f
b) %d
c) %c
d) %s
```

37. Which format specifier is used to read a string using scanf()?

```
a) %f
b) %d
c) %c
d) %s
```

38. What does the \n escape sequence do?

```
a) Creates a tab space
b) Creates a new line
c) Deletes a character
d) Moves cursor back
```

39. How do you read a character using scanf()?

```
a) scanf("%c", &ch);
b) scanf("%c", ch);
c) scanf("&c", ch);
d) scanf("%char", &ch);
```

40. What is the correct way to read two integers in one scanf() call?

```
a) scanf("%d%d", &a, &b);
b) scanf("%d %d", &a, &b);
c) scanf("%d,%d", &a, &b);
d) All of the above
```

Control Structures - Decision Making

41. Which keyword is used for conditional execution in C?

```
a) loop
b) if
c) while
d) for
```

42. What is the correct syntax for a simple if statement?

```
a) if condition { }
b) if (condition) { }
c) if {condition} ( )
d) if [condition] { }
```

43. Which statement provides an alternative block of code when an if condition is false?

```
a) if-else
b) else-if
c) switch
d) continue
```

44. What is the correct syntax for if-else statement?

```
a) if (condition) { } else { }
b) if condition { } else { }
c) if {condition} else { }
d) if (condition) else { }
```

45. Which construct is used to check multiple conditions in sequence?

```
a) Multiple if statements
b) if-else if-else ladder
```

```
c) Nested if
d) All of the above
```

46. What is the purpose of the switch statement?

```
a) To loop through values
b) To compare a variable against multiple values
c) To declare variables
d) To define functions
```

47. Which keyword is used to exit a switch case?

```
a) exit
b) stop
c) break
d) continue
```

48. What happens if you forget to use break in a switch case?

```
a) Program crashes
b) Fall-through to next case
c) Case is skipped
d) Nothing happens
```

49. Which of the following is NOT a valid data type for switch statement?

```
a) int
b) char
c) float
d) enum
```

50. What is the purpose of the default case in switch statement?

```
a) To handle all unmatched cases
b) To start the switch
c) To end the switch
d) To repeat the switch
```

Control Structures - Loops

51. Which of the following is NOT a loop structure in C?

```
a) for
b) while
c) do-while
d) repeat
```

52. What is the correct syntax for a for loop?

```
a) for (init; condition; update) { }
b) for (init, condition, update) { }
c) for {init; condition; update}
d) for [init; condition; update]
```

53. In a for loop, which part is executed only once?

```
a) Condition
b) Initialization
c) Update
```

```
d) Body
```

54. In a for loop, which part is checked before each iteration?

```
a) Initialization
b) Condition
c) Update
d) Body
```

55. What is the correct syntax for a while loop?

```
a) while (condition) { }
b) while condition { }
c) while {condition}
d) while [condition]
```

56. Which loop guarantees at least one execution?

```
a) for
b) while
c) do-while
d) All loops
```

57. What is the correct syntax for a do-while loop?

```
a) do { } while (condition);
b) do (condition) { }
c) do { } while condition;
d) do [condition] { }
```

58. What happens if the condition in a while loop is false from the beginning?

```
a) Loop executes once
b) Loop executes twice
c) Loop doesn't execute at all
d) Program crashes
```

59. What is an infinite loop?

```
a) A loop that executes 1000 times
b) A loop that executes until manually stopped
c) A loop that executes exactly 10 times
d) A loop that executes zero times
```

60. What is the purpose of the break statement in loops?

```
a) To skip current iteration
b) To exit the loop immediately
c) To restart the loop
d) To pause the loop
```

Functions

61. What is a function in C programming?

```
a) A variable
b) A self-contained block of code that performs a specific
task
c) A loop
d) A data type
```

62. What is the purpose of the return statement in a function?

```
a) To restart the function
b) To send a value back to the calling code
c) To delete the function
d) To pause the function
```

63. What does void mean in function declaration?

```
a) The function returns an integer
b) The function returns nothing
c) The function is empty
d) The function has an error
```

64. What is the correct syntax for a function that takes two integers and returns an integer?

```
a) int func(int a, int b)
b) func int(int a, int b)
c) int func(a int, b int)
d) func(int a, int b) int
```

65. What is a function call?

```
a) Declaring a function
b) Executing/invoking a function
c) Defining a function
d) Deleting a function
```

66. What are parameters in a function?

```
a) Variables declared inside the function
b) Input values passed to a function
c) Return values from a function
d) Global variables
```

67. What is the difference between actual parameters and formal parameters?

```
a) Actual parameters are in function call, formal in function definition
b) Formal parameters are in function call, actual in function definition
c) No difference
d) Actual parameters are local, formal are global
```

68. What is function prototyping?

```
a) Writing function code
b) Declaring function before main()
c) Calling a function
d) Defining function parameters
```

69. What is the purpose of function prototyping?

```
a) To make code look neat
b) To allow calling function before defining it
c) To reduce memory usage
d) To increase execution speed
```

70. What is recursion in functions?

```
a) Calling another function
b) Calling the same function within itself
c) Returning multiple values
d) Passing arrays to functions
```

Arrays

71. What is an array in C?

```
a) A single variable
b) A collection of elements of same data type
c) A function
d) A loop
```

72. How do you declare an array of 10 integers?

```
a) int arr[10];
b) int arr(10);
c) int arr{10};
d) int arr[10];
```

73. What is the index of the first element in an array?

```
a) 1
b) 0
c) -1
d) Depends on array size
```

74. How do you access the 5th element of an array arr?

```
a) arr[5]
b) arr(5)
c) arr[4]
d) arr{5}
```

75. What happens if you try to access arr[10] for an array declared as int arr[10]?

```
a) Returns last element
b) Returns first element
c) Causes undefined behavior
d) Program terminates
```

76. How do you initialize an array at the time of declaration?

```
a) int arr[5] = {1,2,3,4,5};
b) int arr[5] = (1,2,3,4,5);
c) int arr[5] = {1 2 3 4 5};
d) int arr[5] = [1,2,3,4,5];
```

77. What is a 2D array?

```
a) An array with only one element
b) An array of arrays (rows and columns)
c) An array with decimal indices
d) An array that can store strings
```

78. How do you declare a 3x3 matrix?

```
a) int matrix[3][3];
b) int matrix(3,3);
```

```
c) int matrix{3}{3};
d) int matrix[3,3];
```

79. How do you access element in 2nd row, 3rd column of matrix?

```
a) matrix[2][3]
b) matrix(2,3)
c) matrix[1][2]
d) matrix{2}{3}
```

80. What is the size of int arr[10] in bytes (assuming int is 4 bytes)?

```
a) 10
b) 40
c) 4
d) 14
```

Strings

81. How are strings represented in C?

```
a) Using string data type
b) Using character array with null terminator
c) Using special string functions only
d) Using double data type
```

82. What is the null terminator in C strings?

```
a) \0
```

```
b) NULL
c) 0
d) \\0
```

83. How do you declare a string to store "Hello"?

```
a) char str[6] = "Hello";
b) char str[5] = "Hello";
c) char str[] = "Hello";
d) Both a and c
```

84. Which header file is needed for string functions like strlen()?

```
a) stdio.h
b) stdlib.h
c) string.h
d) math.h
```

85. What does strlen() function do?

```
a) Copies a string
b) Concatenates strings
c) Returns length of string
d) Compares strings
```

86. What does strcpy() function do?

```
a) Compares strings
b) Copies one string to another
c) Copies one string to another
```

```
d) Returns string length
```

87. What does strcat() function do?

```
a) Compares strings
b) Copies strings
c) Concatenates (joins) two strings
d) Finds string length
```

88. What does strcmp() function do?

```
a) Copies strings
b) Concatenates strings
c) Compares two strings
d) Finds string length
```

89. What is the result of strcmp("abc", "abc")?

```
a) 1
b) 0
c) -1
d) Undefined
```

90. What is the result of strcmp("abc", "xyz")?

```
a) Positive value
b) 0
c) Negative value
d) Undefined
```

Miscellaneous Concepts

91. What is an algorithm?

```
a) A programming language
b) A step-by-step procedure to solve a problem
c) A type of variable
d) A loop structure
```

92. What is a flowchart?

```
a) A type of function
b) A visual representation of an algorithm
c) A data type
d) A programming construct
```

93. Which escape sequence creates a new line?

```
a) \t
b) \n
c) \r
d) \b
```

94. Which escape sequence creates a tab space?

```
a) \t
b) \n
c) \r
d) \b
```

95. What is the purpose of #include <stdio.h>?

```
a) To define variables
b) To include standard input/output library functions
c) To create loops
d) To declare functions
```

96. What is debugging?

```
a) Writing code
b) Finding and fixing errors in code
c) Compiling code
d) Running code
```

97. What is a syntax error?

```
a) Logical mistake in program
b) Error in program logic
c) Error in program structure/rules
d) Runtime error
```

98. What is a logical error?

```
a) Grammar mistake in code
b) Error that causes program to crash
c) Error in program logic that produces wrong output
d) Missing header files
```

99. What is the purpose of comments in C?

```
a) To execute code
b) To explain code to humans
```

```
c) To compile faster
d) To reduce memory usage
```

100. Which symbol is used for single-line comments in C?

```
a) //
b) /*
c) #
d) ;
```

C Programming Assignments & Practice Problems

All programs extracted from the HTML tutorial files with exact titles

LEVEL 1: INTRODUCTION PROGRAMS

Example 1: Working with Numbers Write a program to demonstrate different data types with numbers.

Example 2: Working with Characters Write a program to work with character variables.

Example 3: Simple Student Information Write a program to display student information using multiple data types.

Example 4: Reading an Integer Write a program to read an integer from user input.

Example 5: Reading Multiple Inputs Write a program to read multiple inputs (age, marks, height) from the user.

Example 6: Story without user input Write a program to create a story using predefined variables.

Example 7: Story with user input Write a program to create an interactive story using user input.

LEVEL 2: BASIC OPERATIONS

Example 1: Basic Addition Write a program to perform basic addition.

Example 2: Basic Subtraction Write a program to perform basic subtraction.

Example 3: Basic Multiplication Write a program to perform basic multiplication.

Example 4: Addition with Variables Write a program to add numbers using variables.

Example 5: All Basic Operations Write a program to demonstrate all basic arithmetic operations.

Example 6: Add Two Numbers from User Write a program to add two numbers entered by the user.

Example 7: Understanding Remainder (%) Write a program to demonstrate the modulus operator.

Example 8: Is One Number Bigger? Write a program to compare two numbers.

Example 9: Ask for User's Age Write a program to ask and display user's age.

Example 10: Simple Personal Information Write a program to collect and display personal information.

Example 11: Is a Number Big? Write a program to check if a number is greater than a certain value.

Example 12: Can You Vote? Write a program to check voting eligibility.

Example 13: Is Number Even or Odd? Write a program to check if a number is even or odd.

Practice 14: Grade Calculator (Simple) Write a program to calculate grades based on marks.

Practice 15: Simple Calculator Write a program to create a simple calculator.

Example 16: Find the Biggest of Two Numbers Write a program to find the larger of two numbers.

Example 17: Age Group Checker Write a program to classify age groups.

LEVEL 3: CONTROL STRUCTURES

Example 1: Count from 1 to 5 Write a program to count from 1 to 5 using loops.

Example 2: Count from 1 to 10 Write a program to count from 1 to 10.

Example 3: Count Backwards Write a program to count backwards.

Example 4: Print User's Name Multiple Times Write a program to print user's name multiple times.

Example 5: Addition Table Write a program to create an addition table.

Example 6: Sum of First N Numbers Write a program to calculate sum of first N numbers.

Example 7: Even Numbers from 2 to 20 Write a program to print even numbers from 2 to 20.

Example 8: Check if Number is Positive Write a program to check if a number is positive.

Example 9: Check if One Number is Greater Than Another Write a program to compare two numbers.

Example 10: Check Voting Eligibility Based on Age Write a program to check voting eligibility.

Example 11: Check Pass or Fail Write a program to determine pass or fail.

Example 12: Check if Student Has Passed or Failed Based on Marks Write a program for student pass/fail evaluation.

Example 13: Check if Number is Positive or Negative Write a program to check number sign.

Example 14: Check if Number is Odd or Even Write a program to check odd or even.

Example 15: Check if a Year is a Leap Year Write a program to check leap year.

Example 16: Positive, Negative, or Zero Check Write a program for three-way number check.

Example 17: Print Day of Week Based on User's Choice Write a program to display day names.

Example 18: Print Month Based on User's Choice Write a program to display month names.

Example 19: Simple Age Groups Write a program to categorize age groups.

Example 20: Leap Year Checker Write a program to check leap year with complete logic.

Example 21: Voting Eligibility Write a program to check voting eligibility.

Example 22: Vowel or Consonant Write a program to check vowel or consonant.

Example 23: Positive, Negative, or Zero Write a program for number classification.

Example 24: Counting from 1 to 5 Write a program using while loop to count.

Example 25: Count Number of Digits in a Number Write a program to count digits.

Example 26: Counting from 1 to 5 with a for loop Write a program using for loop.

Example 27: A simple menu Write a program to create a simple menu.

Example 28: Simple Calculator for Two Numbers Write a calculator program.

Example 29: Find Maximum of Three Numbers Write a program to find maximum of three numbers.

Example 30: Print Numbers from 1 to n Write a program to print numbers from 1 to n.

Example 31: Print Numbers from n to 1 Write a program to print numbers in reverse.

Example 32: Print Odd Numbers Up to n Write a program to print odd numbers.

Example 33: Multiplication Table Write a program to print multiplication table.

Example 34: Sum of Even Numbers Write a program to calculate sum of even numbers.

Example 35: Simple Grade Calculator Write a program for grade calculation.

Example 36: Find the Bigger Number Write a program to find the bigger number.

Example 37: Simple Number Guessing Write a simple number guessing game.

LEVEL 4: FUNCTIONS & ARRAYS

Example 1: Simple Function Example Write a program with a simple function.

Example 2: A function that adds two numbers Write a program with an addition function.

Example 3: Area of a Rectangle Write a program to calculate rectangle area using functions.

Example 4: Array Declaration and Initialization Write a program to declare and initialize arrays.

Example 5: Accessing and printing elements Write a program to access array elements.

Example 6: Printing all elements with a loop Write a program to print all array elements.

Example 7: Temperature Converter Write a program to convert temperatures.

Example 8: Prime Number Checker Write a program to check prime numbers.

Example 9: Factorial Function Write a program to calculate factorial using functions.

Example 10: Sum of an Array Write a program to calculate sum of array elements.

Example 11: Search Element in Array Write a program to search for an element in array.

Example 12: Reverse an Array Write a program to reverse array elements.

Example 13: Count Frequency of Elements Write a program to count element frequency in array.

Example 14: Basic String Operations Write a program for basic string operations.

Example 15: Using string functions Write a program using string library functions.

Example 16: Simple Username Greeting Write a program for username greeting.

Example 17: String Length Without strlen() Write a program to find string length without library function.

Example 18: Reverse a String Write a program to reverse a string.

Example 19: Count Vowels and Consonants Write a program to count vowels and consonants.

Example 20: Check Palindrome Write a program to check if string is palindrome.

Example 21: Simple Word Count Write a program to count words in a string.

LEVEL 5: ADVANCED PROGRAMS

Example 1: Simple Calculator Write a program for an advanced calculator.

Example 2: Simple Number Game Write a program for a number guessing game.

1. Write a program to check whether a person is eligible to vote or not.

2. Write a program to check whether a character is vowel or consonant.

3. Write a program to check whether a number is positive, negative or zero.

4. Write a program to calculate the grade of a student.

5. Write a program to check whether a character is vowel or not.

6. Write a program to print your name 10 times using while loop.

8. Write a program to add two numbers (10, 15).

9. Write a program to multiply three numbers (2, 3, 5).

10. Write a program to find the square of a given number (7).

11. Write a program to find the area of a rectangle (l=5, b=4).

12. Write a program to calculate the perimeter of a circle.

13. Write a program to add two numbers (10, 20).

14. Write a program to find the square and cube of a number.

15. Write a program to find the sum, product, and average of three numbers (2, 5, 7).

16. Write a program to accept three integer numbers from the user and calculate the sum, product, and average.

17. Write a program to find the area of a square.

18. Write a program to find the area of a rectangle.

19. Write a program to swap the values of two numbers.

Answer Key

1. b) Dennis Ritchie
2. a) 1969-1973
3. c) To serve as the entry point of the program
4. c) Automatic garbage collection
5. a) Integrated Development Environment
6. b) C-Free
7. c) .c
8. b) Successful program termination
9. b) A storage location with a name that holds data
10. c) char
11. c) %d
12. c) 4
13. c) float
14. b) %f
15. a) Displays 2 decimal places
16. b) char ch = 'A';
17. b) -128 to 127
18. c) double
19. b) 8
20. c) %lf
21. a) +
22. b) *
23. b) /
24. b) Modulus (remainder)
25. b) 1
26. b) 3
27. b) ==
28. b) !=

29. a) 1
30. b) False
31. a) 1
32. a) 1
33. b) printf()
34. b) scanf()
35. b) To get the address of a variable
36. b) %d
37. d) %s
38. b) Creates a new line
39. a) scanf("%c", &ch);
40. d) All of the above
41. b) if
42. b) if (condition) { }
43. a) if-else
44. a) if (condition) { } else { }
45. b) if-else if-else ladder
46. b) To compare a variable against multiple values
47. c) break
48. b) Fall-through to next case
49. c) float
50. a) To handle all unmatched cases
51. d) repeat
52. a) for (init; condition; update) { }
53. b) Initialization
54. b) Condition
55. a) while (condition) { }
56. c) do-while
57. a) do { } while (condition);
58. c) Loop doesn't execute at all
59. b) A loop that executes until manually stopped
60. b) To exit the loop immediately
61. b) A self-contained block of code that performs a specific task
62. b) To send a value back to the calling code
63. b) The function returns nothing
64. a) int func(int a, int b)
65. b) Executing/invoking a function

66. b) Input values passed to a function
67. a) Actual parameters are in function call, formal in function definition
68. b) Declaring function before main()
69. b) To allow calling function before defining it
70. b) Calling the same function within itself
71. b) A collection of elements of same data type
72. a) int arr[10];
73. b) 0
74. c) arr[4]
75. c) Causes undefined behavior
76. a) int arr[5] = {1,2,3,4,5};
77. b) An array of arrays (rows and columns)
78. a) int matrix[3][3];
79. c) matrix[1][2]
80. b) 40
81. b) Using character array with null terminator
82. a) \0
83. d) Both a and c
84. c) string.h
85. c) Returns length of string
86. c) Copies one string to another
87. c) Concatenates (joins) two strings
88. c) Compares two strings
89. b) 0
90. c) Negative value
91. b) A step-by-step procedure to solve a problem
92. b) A visual representation of an algorithm
93. b) \n
94. a) \t
95. b) To include standard input/output library functions
96. b) Finding and fixing errors in code
97. c) Error in program structure/rules
98. c) Error in program logic that produces wrong output
99. b) To explain code to humans
100. a) //