



Certified C Programmer (CCLP)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working C programmer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1

Domain 1: C Fundamentals and Syntax

[CONFIG](#)

Sarah Miller compiles this program and checks what it prints.

```
#include <stdio.h>
int main(void) {
    int x = 5, y = 2;
    printf("%d", x / y);
    return 0;
}
```

Listing 1.1 Integer division.

What does the program print?

- ☐ A 2
- ☐ B 2.5
- ☐ C 3
- ☐ D 2.50

Correct answer: A

Both operands are of type int, so the division is integer division and the fractional part is dropped, leaving 2. Floating point results such as 2.5 or 2.50 would require a float or double operand.

Question 2

Domain 2: Data Types and Operators

DATA

James Carter evaluates each bitwise expression, with one result missing.

Expression	Result
5 & 3	1
5 2	7
5 ^ 1	(to choose)
1 << 3	8

Table 2.1 Bitwise expressions.

What is the value of 5 ^ 1?

- ☐ A 6
- ☐ B 4
- ☐ C 2
- ☐ D 1

Correct answer: B

The exclusive or operator flips only the bits that differ, so 101 xor 001 gives 100, which is 4. The bitwise and, or, and shift operators produce the other rows and do not apply here.

Question 3

Domain 3: Control Flow and Functions

CONFIG

Emma Wilson traces this recursive function called as f(4).

```
int f(int n) {  
    if (n <= 1) return 1;  
    return n * f(n - 1);  
}
```

Listing 3.1 A recursive function.

What does f(4) return?

- ☐ A 12
- ☐ B 16
- ☐ C 24
- ☐ D 10

Correct answer: C

The function multiplies n by f(n - 1) until the base case, so it computes 4 times 3 times 2 times 1, which is 24. That is the factorial of 4, and the other values do not match the product.

Question 4

Domain 4: Pointers, Memory and Structs

CONFIG

David Brown steps through this pointer code and checks the final value of a.

```
int a = 10;
int *p = &a;
*p = *p + 5;
printf("%d", a);
```

Listing 4.1 Writing through a pointer.

What is printed for a?

- ☐ A 10
- ☐ B 5
- ☐ C 20
- ☐ D 15

Correct answer: D

The pointer p holds the address of a, so writing through *p updates a itself, giving 10 plus 5, which is 15. The value 10 would print only if the assignment through the pointer had no effect.

Question 5

Domain 5: Arrays, Strings and the Standard Library

DATA

Olivia Davis maps each string function to its purpose, with one purpose missing.

Function	Purpose
strlen	Length of string
strcpy	Copy a string
strcmp	(to choose)
strcat	Join two strings

Table 5.1 String library functions.

What does strcmp do?

- ☐ A Reverse a string
- ☐ B Compare strings
- ☐ C Count the words
- ☐ D Split a string

Correct answer: B

The strcmp function compares two strings and returns a value that is zero when they are equal. It does not modify either string, and reversing, counting, or splitting are not part of what it does.

Question 6

Domain 6: Preprocessor, Compilation and Best Practices

CONFIG

Liam Anderson defines a macro without parentheses and uses it with a sum.

```
#define SQUARE(x) x * x
int r = SQUARE(2 + 3);
```

Listing 6.1 A macro with no parentheses.

What value does r hold?

- ☐ A 25
- ☐ B 10
- ☐ C 11
- ☐ D 13

Correct answer: C

The preprocessor expands the macro literally to `2 + 3 * 2 + 3`, and precedence makes that 2 plus 6 plus 3, which is 11. Wrapping the parameter in parentheses would give the expected 25.

Question 7

Domain 1: C Fundamentals and Syntax

A team calls `printf` but the compiler warns about an implicit declaration because a header is missing. Which header declares `printf`?

- ☐ A `<stdlib.h>`
- ☐ B `<string.h>`
- ☐ C `<stdio.h>`
- ☐ D `<math.h>`

Correct answer: C

The standard input and output functions such as `printf` are declared in `stdio.h`, so including it clears the implicit declaration warning. The other headers cover general utilities, string handling, and math functions.

Question 8

Domain 3: Control Flow and Functions

A helper function must keep a counter that survives across calls without exposing it as a global. Which storage keyword fits?

- ☐ A `extern`
- ☐ B `static`
- ☐ C `const`
- ☐ D `auto`

Correct answer: B

A static local variable keeps its value between calls while staying private to the function, which is exactly what the counter needs. The extern keyword links to an outside definition, const blocks changes, and auto is the default that does not persist.

Question 9

Domain 4: Pointers, Memory and Structs

A program calls malloc inside a loop but never frees the blocks, and its memory use keeps climbing until it runs low. What is this defect?

- ☐ A Stack overflow
- ☐ B Dangling read
- ☐ C Null deref
- ☐ D Memory leak

Correct answer: D

Allocating memory without ever freeing it is a memory leak, so usage grows without bound as the loop runs. A stack overflow comes from too much recursion, and the other options describe unrelated pointer faults.

Question 10

Domain 5: Arrays, Strings and the Standard Library

A loop writes to array indices from 0 up to and including the array length, one slot past the last valid element. What defect is this?

- ☐ A Integer overflow
- ☐ B Buffer overrun
- ☐ C Type mismatch
- ☐ D Syntax mistake

Correct answer: B

Writing one slot past the last valid index is a buffer overrun that corrupts memory next to the array and is undefined behavior. The loop should stop at length minus one, and the other options do not describe an out of bounds write.