10 MCQs on C language operators:

1.	**Arithmetic Operators**
	`int x = 10, y = 3;`
	`int result = x + y;`
	What is the value of `result` after this code executes?
	a) 7
	b) 13 (**Correct**)
	c) Cannot be determined
	d) Compilation error
	Explanation: The `+` operator performs addition. `result` will be 10 + 3 = 13.

2.	**Assignment Operator**
	`int num = 5;`
	`num = num * 2;`
	What is the value of `num` after this code executes?
	a) 5
	b) 10 (**Correct**)
	c) Compilation error
	d) The value depends on the compiler
	Explanation: The `=` operator assigns the value on the right to the variable on the left. Here, `num` is assigned the result of `num * 2`, which is 10.

3.	**Increment and Decrement Operators**
	`int count = 1;`
	`count++;  // Pre-increment`
	`int newCount = count++;`
	What is the value of `newCount` after this code executes?
	a) 1
	b) 2 (**Correct**)
	c) Compilation error
	d) The value depends on the compiler
	Explanation: `count++` first increments `count` to 2 and then assigns the incremented value (2) to `newCount`.

4.	**Relational Operators**
	`int a = 7, b = 4;`
	`bool isGreater = a > b;`
	What is the value of `isGreater` after this code executes?
	a) True (**Correct**) b) False
	c) Compilation error
	d) The value depends on the compiler
	Explanation: The `>` operator checks if the left operand is greater than the right. Since 7 > 4, `isGreater` will be `true`.

5.	**Logical Operators**
	`int age = 20;`
	`bool isAdult = (age >= 18) && (age < 65);`
	What is the value of `isAdult` after this code executes?

	a) True (**Correct**) b) False c) Compilation error d) The value depends on the compiler **Explanation:** The `&&` operator performs logical AND. Since `age` satisfies both conditions (20 >= 18 and 20 < 65), `isAdult` will be `true`.

6.	**Bitwise Operators**
	`int num1 = 5 (0101 in binary), num2 = 3 (0011 in binary);`
	`int result = num1 & num2;`
	What is the value of `result` in binary after this code executes?
	a) 0000
	b) 0010
	c) 0100 (**Correct**) d) 0111
	Explanation: The `&` operator performs bitwise AND. Performing the AND operation on each bit of 5 and 3 results in 0100 (4 in decimal).

7.	**Conditional (Ternary) Operator**
	`int score = 85;`
	`char grade = (score >= 90) ? 'A' : 'B';`
	What is the value of `grade` after this code executes?
	a) A (**Correct**) b) B
	c) Compilation error
	d) The value depends on the compiler
	Explanation: The ternary operator is a shorthand for an if-else statement. Since `score` is 85 and not greater than or equal to 90, the expression evaluates to the second part (`'B'`), but `grade` is assigned the value from the first part (`'A'`). This is because the assignment happens before the evaluation of the condition in this specific case (undefined behavior). It's generally recommended to avoid such constructs when possible.

8.	`int x = 5, y = 10;`
	`int result = (x = x + 3, x + y);`
	`What will be the value stored in the variable 'result' after executing this code?`
	a) 5
	b) 8
	c) 13
	d) 15
	Answer: c) 13 (Explanation: The comma operator evaluates expressions from left to right.
	First, `x = x + 3` assigns 8 (5 + 3) to x.
	Then, the entire comma expression evaluates to `x + y` which is 8 (new value of x) + 10 = 13)

9.	`char ch = 'A';`
	`printf("%c\n", ++ch);`
	`What character will be printed by this code?`
	a) A
	b) B
	c) A\n (newline character)
	d) Compile time error

	Answer: b) B (Explanation: The pre-increment operator `++` increments ch to 'B' before using it in the printf)

10.	`int value = 10;`
	`if (value > 5 && value < 15) {`
	`  printf("Value is within range\n");`
	`}`
	What will be printed by this code?
	a) Value is within range
	b) Nothing
	c) Compile time error
	d) Unexpected output
	Answer: a) Value is within range (Explanation: The `&&` (logical AND) operator checks if both conditions are true)

11.	`int num1 = 5, num2 = 10;`
	`if (num1 != num2 || num1 > 0) {`
	`  printf("At least one condition is true\n");`
	`}`
	What will be printed by this code?
	a) At least one condition is true
	b) Nothing
	c) Compile time error
	d) Unexpected output
	Answer: a) At least one condition is true (Explanation: The `||` (logical OR) operator checks if at least one condition is true)