C Programming- Functions (Answers)

1. What is the scope of a variable declared inside a function in C?

A) The variable is accessible throughout the entire program.
B) The variable is accessible only within the function where it is declared.
C) The variable is accessible only within the main () function.
D) The variable is accessible globally across all functions.

Answer: B) The variable is accessible only within the function where it is declared.
Explanation: A local variable's scope is limited to the function in which it is declared and is not accessible outside of that function.

2. What will be the output of the following code?

```
#include <stdio.h>
void func (int x) {
   if (x == 0) return;
   printf ("%d ", x);
   func (x - 1);
}
int main () {
   func (3);
   return 0;
}
```

A) 3 2 1
B) 1 2 3
C) 3 2 1 0
D) 0 1 2 3

Answer: A) 3 2 1
Explanation: The function prints the value of x and then recursively calls itself with x - 1 until x reaches 0, at which point the recursion stops.

3. What will be the output of the following code?

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

```
void modify (int x) {
   x = 100;
}
int main () {
   int num = 10;
   modify(num);
   printf ("%d", num);
   return 0;
}
```

A) 100
B) 10
C) 0
D) Undefined behavior

Answer: B) 10
Explanation: This is pass by value, meaning the modify () function only changes the local copy of num, not the original variable in main (). Hence, the output will be the original value of num, which is 10.

4. What is the role of the return statement in a function?

A) It ends the execution of the program immediately.
B) It stops the function's execution and returns control to the calling function, optionally returning a value.
C) It temporarily suspends the execution of the program.
D) It defines the return type of a function.

Answer: B) It stops the function's execution and returns control to the calling function, optionally returning a value.
Explanation: The return statement terminates the function execution and sends control back to the calling function. If the function has a return type, it can return a value.

5. Which of the following statements is correct regarding library functions in C?

A) Library functions are defined in the program and do not need to be included explicitly.
B) Library functions are built into the C language and are always available without inclusion.
C) Library functions must be included explicitly by including the correct header files.
D) Library functions can only be used inside the main() function.

Answer: C) Library functions must be included explicitly by including the correct header files.
Explanation: Library functions are defined in standard header files like <stdio.h>, <stdlib.h>, etc., and must be included in the program to be used.

6. What is the output of the following code?

```
#include <stdio.h>
int func(int a) {
   if (a == 0) return 1;
   return a * func(a - 1);
}
int main() {
   printf("%d", func(5));
   return 0;
}
```

A) 5
B) 120
C) 24
D) 1

Answer: B) 120
Explanation: The function func is a recursive function calculating the factorial of the number 5. The recursion calculates 5 * 4 * 3 * 2 * 1, which equals 120.

7. What is the output of the following program?

```
#include <stdio.h>
void func(int *a, int *b) {
   *a = *a + *b;
}
int main() {
   int x = 5, y = 10;
   func(&x, &y);
```

```
    printf("%d %d", x, y);
    return 0;
}
```

A) 15 10
B) 5 15
C) 10 10
D) 5 5

Answer: A) 15 10
Explanation: The function func modifies the value of x by adding y to it. So, after calling func, x becomes 15 and y remains 10. The output will be 15 10.

8. What is the purpose of a function prototype in C?

A) It defines the function's body.
B) It specifies the function's return type and parameter types before the function is defined.
C) It is only needed if a function is declared after the main() function.
D) It is used to define default values for function arguments.

Answer: B) It specifies the function's return type and parameter types before the function is defined.
Explanation: A function prototype provides the compiler with information about the function's return type and parameters before its actual definition, allowing you to use the function before it is fully defined in the code.

9. Which of the following is correct regarding the void return type in C?

A) The function must return NULL.
B) The function can return a value of any type.
C) The function does not return any value.
D) The function can only be called from main.

Answer: C) The function does not return any value.
Explanation: In C, a void function does not return any value, meaning no data is passed back to the caller.

10. What is the output of the following code?

```
#include <stdio.h>
void func(int a, int b) {
    a = a + 5;
```

```
    b = b - 5;
    printf("%d %d", a, b);
}
int main() {
    int x = 10, y = 20;
    func(x, y);
    printf("%d %d", x, y);
    return 0;
}
```

A) 15 15 10 20
B) 15 15 10 20
C) 15 15 15 15
D) 10 20 10 20

Answer: A) 15 15 10 20
Explanation: The function func changes the values of a and b locally (by value), so these modifications do not affect the variables x and y in main. The output of func will be 15 15, but the values of x and y remain 10 and 20 after the function call.

11. If a function prototype is defined as: **int calculate(int x, int y);**

What will happen if the function is called with: **calculate(10.5, 20.2);**

A) The program will run correctly, converting the float values to int.
B) Compilation error due to type mismatch.
C) The program will run, but the values will be truncated.
D) T The program will generate a warning.

Answer: B) Compilation error due to type mismatch.
Explanation: The function calculate(int, int) expects integer values, but 10.5 and 20.2 are floating-point numbers. This will lead to a compilation error because the argument types do not match the expected int type.

12. What is the main purpose of the free() function in C?

A) To allocate memory for a program
B) To deallocate memory that was previously allocated
C) To initialize a pointer to NULL
D) To dynamically allocate memory during runtime

Answer: B) To deallocate memory that was previously allocated
Explanation: The free() function is used to release dynamically allocated memory, making it available for future use.

13. What is the output of the following program?

```
#include <stdio.h>
void func(int* a) {
   (*a)++;
}
int main() {
   int a = 5;
   func(&a);
   func(&a);
   printf("%d", a);
   return 0;
}
```

A) 7
B) 6
C) 5
D) 8

Answer: A) 7
Explanation: The function func() is called twice, each time incrementing a by 1, so the final value of a is 7.

14. What does a function pointer store in C?

A) The name of the function.
B) The return value of the function.
C) The address of the function.
D) The arguments of the function.

Answer: C) The address of the function.
Explanation: A function pointer holds the address of a function so that it can be called dynamically.

15. What does the static keyword do when applied to a function in C?

A) Makes the function accessible from all files.
B) Keeps the function private to its own file.
C) Makes the function run faster.
D) Makes the function run only once.

Answer: B) Keeps the function private to its own file.
Explanation: The static keyword makes the function only accessible within the file where it is defined, preventing it from being accessed by other files.

C Programming MCQ Questions and Answers

16.

What is the output of the following code?

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

A) 4

B) 5

C) 6

D) Compilation Error

Answer: C

Explanation: ++x increments x by 1 before using its value, so the output is 6.

17.

Which of the following is not a valid storage class in C?

A) auto

B) extern

C) register

D) mutable

Answer: D

Explanation: 'mutable' is not a valid storage class in C; it is used in C++.

18.

What is the output of the following code?

```
#include <stdio.h>
void func() {
    static int x = 0;
    x++;
    printf("%d ", x);
}
int main() {
    func();
    func();
    return 0;
}
```

A) 1 1

B) 1 2

C) 2 2

D) Compilation Error

Answer: B

Explanation: Static variable 'x' retains its value between function calls, so output is 1 2.

19.

Which of the following function calls is correct?

A) printf("%d", 10);

B) printf("%f", 10);

C) printf("%c", "A");

D) printf(10);

Answer: A

Explanation: '%d' matches with the integer value 10; other options have type mismatches.

20.

Which of these is not a valid way to define a function in C?

A) int func();

B) void func(int x);

C) int func(int x,);

D) void func();

Answer: C

Explanation: Trailing comma in 'int func(int x,);' is a syntax error.

21.

What does the `extern` keyword indicate?

A) Variable is local

B) Variable is global

C) Variable has a fixed value

D) Variable is declared in another file

Answer: D

Explanation: 'extern' indicates that the variable is declared in another file.

22.

How is a function prototype defined in C?

A) By specifying return type and function name only

B) By specifying return type, function name, and parameters

C) By specifying function body only

D) By specifying only the function name

Answer: B

Explanation: A function prototype specifies the return type, function name, and parameters.

23.

What happens when a `return` statement is used without any expression?

A) Compilation error

B) Program exits

C) Control returns to the calling function

D) Undefined behavior

Answer: C

Explanation: 'return' without an expression transfers control to the calling function.

24.

What is the output of the following code?

```
#include <stdio.h>
int func(int x, int y) {
    return x + y;
}
int main() {
    printf("%d", func(3, 4));
    return 0;
}
```

A) 3

B) 4

C) 7

D) Compilation Error

Answer: C

Explanation: The function adds 3 and 4, returning 7.

25.

Which of the following correctly passes an array to a function?

A) func(arr[]);

B) func(&arr);

C) func(arr);

D) func(*arr);

Answer: C

Explanation: 'func(arr);' passes the base address of the array to the function.

26.

What is the return type of the `main` function in C?

A) void

B) int

C) float

D) char

Answer: B

Explanation: The return type of the 'main' function in C is 'int'.

27.

What does a function declaration specify?

A) The function's name

B) The return type and parameters of the function

C) The function's body

D) All of the above

Answer: B

Explanation: A function declaration specifies the return type and parameters.

28.

What is the output of the following code?

#include <stdio.h>

```
int main() {
    printf("%d", sizeof(int));
    return 0;
}
```

A) 2

B) 4

C) 8

D) Compiler dependent

Answer: D

Explanation: The size of 'int' is compiler dependent, commonly 2, 4, or 8 bytes.

29.

What is the correct syntax to return a pointer from a function?

A) int *func();

B) int func*();

C) int &func();

D) int func*[];

Answer: A

Explanation: 'int *func();' correctly specifies a function returning a pointer to int.

30.

What is the purpose of a `void` function?

A) It performs a task but does not return a value

B) It returns an integer

C) It is used for memory allocation

D) It must have a return statement

Answer: A

Explanation: 'void' functions perform tasks but do not return a value.