C Language MCQs

1. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int a = 5, b = 2;
   printf("%d", a++ + ++b);
   return 0;
}
```

a) 7 b) 8 c) 9 d) 10

Answer: b) 8

Explanation:

- a++ is post-increment, so the value of a used in the expression is 5, and then a is incremented to 6.
- ++b is pre-increment, so the value of b used in the expression is 3, and then b is incremented to 3.
- Therefore, the expression evaluates to 5 + 3 = 8.

2. Which of the following is not a valid C variable name?

a) my_variable

b) 123variable

c) variable_name

d) _variable

Answer: b) 123variable

Explanation:

- Variable names in C cannot start with a digit.

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

a) To define a new data type

b) To declare a function

c) To include the standard input/output library

d) To create a new file

Answer: c) To include the standard input/output library

Explanation:

- The #include <stdio.h> directive includes the standard input/output library, which provides functions like printf() and scanf() for input and output operations.

4. What is the output of the following code?

C

```
#include <stdio.h>

int main() {
   int x = 10;
   if (x++ > 10) {
      printf("Hello");
   } else {
      printf("World");
   }
   return 0;
}
```

a) Hello b) World c) Hello World d) No output

Answer: b) World

Explanation:

- The condition x++ > 10 is evaluated to false because the value of x used in the comparison is 10, and then x is incremented to 11.

5. Which loop is most suitable for iterating a specific number of times?

a) while loop

b) do-while loop

c) for loop d) switch-case

Answer: c) for loop

Explanation:

- The for loop is specifically designed for iterating a known number of times. It has three parts: initialization, condition, and increment/decrement.

6. What is the output of the following code?

C

```
#include <stdio.h>

int main() {
```

```
    int i;
    for (i = 0; i < 5; i++) {
        if (i == 3) {
            continue;
        }
        printf("%d ", i);
    }
    return 0;
}
```

a) 0 1 2 3 4

b) 0 1 2 4

c) 0 1 2 3

d) 1 2 3 4

Answer: b) 0 1 2 4

Explanation:

- The continue statement skips the current iteration of the loop and moves to the next iteration.

7. What is the purpose of the break statement in a loop?

a) To skip the current iteration

b) To exit the loop immediately

c) To increment the loop counter

d) To decrement the loop counter

Answer: b) To exit the loop immediately

Explanation:

- The break statement terminates the loop immediately, regardless of the loop condition.

8. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
    int a = 10, b = 5;
    printf("%d", a / b * 2);
```

```
    return 0;
}
```

a) 2 b) 4 c) 5 d) 10

Answer: b) 4

Explanation:

- Integer division is performed first: a / b = 10 / 5 = 2.
- Then, the result is multiplied by 2: 2 * 2 = 4.

9. Which of the following is a valid way to declare a pointer to an integer in C?

a) int *ptr;*

b) int ptr;

c) *int ptr;

d) *int* ptr;

Answer: a) int *ptr;

Explanation:

- The correct syntax for declaring a pointer to an integer is int *ptr;.

10. What is the purpose of the sizeof operator in C?

a) To determine the size of a data type or variable

b) To allocate memory dynamically

c) To perform arithmetic operations

d) To compare two values

Answer: a) To determine the size of a data type or variable

Explanation:

- The sizeof operator returns the size of a data type or variable in bytes.

11. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
    char str[] = "Hello";
    printf("%s", str);
    return 0;
}
```

a) H b) e c) l d) Hello

Answer: d) Hello

Explanation:

- The %s format specifier is used to print a string.

12. Which of the following is not a valid way to initialize an array in C?

a) int arr[5] = {1, 2, 3, 4, 5};

b) int arr[] = {1, 2, 3, 4, 5};

c) int arr[5] = {1, 2, 3};

d) int arr[5] = 1, 2, 3, 4, 5;

Answer: d) int arr[5] = 1, 2, 3, 4, 5;

Explanation:

- The correct syntax for initializing an array is to enclose the values within curly braces {}.

13. What is the purpose of the #define directive in C?

a) To declare a function

b) To define a constant

c) To include a header file

d) To create a new data type

Answer: b) To define a constant

Explanation:

- The #define directive is used to define symbolic constants, which are replaced by their values during preprocessing.

14. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int x = 5;
   if (x > 0) {
      printf("Positive");
   } else if (x < 0) {
      printf("Negative");
   } else {
```

```
    printf("Zero");
  }
  return 0;
}
```

a) Positive b) Negative c) Zero d) No output

Answer: a) Positive

Explanation:

- The value of x is greater than 0, so the first condition is true, and "Positive" is printed.

15. What is the output of the following code?

C

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

a) 2 b) 1 c) 0 d) 3

Answer: b) 1

Explanation:

- The % operator is the modulus operator, which gives the remainder of the division. 5 divided by 2 leaves a remainder of 1.

16. Which of the following is a correct way to declare a function in C?

a) return_type function_name();

b) return_type function_name(parameters);

c) function_name(parameters) return_type;

d) function_name return_type(parameters);

Answer: b) return_type function_name(parameters);

Explanation:

- The correct syntax for declaring a function is return_type function_name(parameters).

18. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int a = 10, b = 5;
   printf("%d", a & b);
   return 0;
}
```

a) 0 b) 1 c) 5 d) 10

Answer: c) 0

Explanation:

- The & operator performs a bitwise AND operation.
- Binary representation of 10 is 1010.
- Binary representation of 5 is 0101.
- **Performing bitwise AND: 1010 & 0101**

0000

- The result in decimal is 0.

17. What is the purpose of the void keyword in C?

a) To indicate that a function does not return any value

b) To declare a variable that can hold any data type

c) To allocate memory dynamically

d) To define a constant

Answer: a) To indicate that a function does not return any value

Explanation:

- The void keyword is used to indicate that a function does not return any value.

19. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int x = 10, y = 5;
```

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

a) x = 5, y = 10

b) x = 10, y = 5

c) x = 15, y = 10

d) x = 10, y = 15

Answer: a) x = 5, y = 10

Explanation:

- This code snippet demonstrates a technique for swapping the values of two variables without using a temporary variable.

20. What is the difference between == and = operators in C?

a) Both are used for assignment.

b) == is used for assignment, and = is used for comparison.

c) = is used for assignment, and == is used for comparison.

d) Both are used for comparison.

Answer: c) = is used for assignment, and == is used for comparison.

Explanation:

- = is the assignment operator, used to assign a value to a variable (e.g., x = 5).
- == is the equality operator, used to compare two values for equality (e.g., if (x == 5) ...).

21. What is the purpose of the struct keyword in C?

a) To define a new data type

b) To declare a function

c) To include a header file

d) To create a new file

Answer: a) To define a new data type

Explanation:

- The struct keyword is used to create a new data type that groups together variables of different data types under a single name.

22. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int arr[] = {10, 20, 30, 40, 50};
   printf("%d", arr[2]);
   return 0;
}
```

a) 10 b) 20 c) 30 d) 40

Answer: c) 30

Explanation:

- Array indices start from 0. So, arr[2] refers to the third element of the array, which is 30.

23. Which of the following is a preprocessor directive in C?

a) int

b) if

c) for

d) #include

Answer: d) #include

Explanation:

- Preprocessor directives are lines that begin with # and are processed before the actual compilation. #include is a common preprocessor directive used to include header files.

24. What is the output of the following code?

C

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

a) 5 b) 7 c) 10 d) 2

Answer: c) 10

Explanation:

- x *= 2 is a shorthand for x = x * 2, which multiplies the value of x by 2 and assigns the result back to x.

25. What is the purpose of the malloc() function in C?

a) To declare a function

b) To allocate memory dynamically

c) To deallocate memory

d) To compare two values

Answer: b) To allocate memory dynamically

Explanation:

- malloc() is a library function used to allocate a block of memory dynamically at runtime.

26. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   char str1[] = "Hello";
   char str2[] = "World";
   printf("%s %s", str1, str2);
   return 0;
}
```

a) Hello b) World c) HelloWorld d) Hello World

Answer: d) Hello World

Explanation:

- The printf() function prints the two strings with a space in between.

27. Which of the following is a correct way to pass an array to a function in C?

a) function_name(int arr[]);

b) function_name(int *arr);

c) function_name(&arr);

d) Both a) and b) are correct.

Answer: d) Both a) and b) are correct.

Explanation:

- Arrays are inherently passed by reference in C. So, both int arr[] and int *arr are valid ways to declare an array parameter in a function.

28. What is the purpose of the typedef keyword in C?

a) To define a new data type

b) To declare a function

c) To include a header file

d) To create a new file

Answer: a) To define a new data type

Explanation:

- The typedef keyword is used to create a new name for an existing data type.

29. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int a = 10;
   int *ptr = &a;
   printf("%d", *ptr);
   return 0;
}
```

a) Address of a b) 10 c) Garbage value d) 0

Answer: b) 10

Explanation:

- ptr is a pointer that stores the address of the variable a.
- *ptr is the dereference operator, which accesses the value stored at the address pointed to by ptr.

30. What is the purpose of the switch-case statement in C?

a) To iterate a specific number of times

b) To execute different blocks of code based on the value of an expression

c) To handle errors

d) To allocate memory dynamically

Answer: b) To execute different blocks of code based on the value of an expression

Explanation:

- The switch-case statement provides a way to execute different blocks of code based on the value of an expression.

31. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int a = 5, b = 2;
   if (a > b) {
      printf("a is greater than b");
   } else {
      printf("b is greater than or equal to a");
   }
   return 0;
}
```

a) a is greater than b

b) b is greater than or equal to a

c) a is equal to b

d) No output

Answer: a) a is greater than b

Explanation:

- The condition a > b is true, so the first block of code within the if statement is executed.

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

a) To exit the function immediately

b) To skip the current iteration of a loop

c) To allocate memory dynamically

d) To define a constant

Answer: a) To exit the function immediately

Explanation:

- The return statement is used to exit a function and optionally return a value to the calling function.

33. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int i, sum = 0;
   for (i = 1; i <= 5; i++) {
      sum += i;
   }
   printf("Sum = %d", sum);
   return 0;
}
```

a) 5 b) 10 c) 15 d) 25

Answer: c) 15

Explanation:

- The code calculates the sum of the first 5 natural numbers (1 + 2 + 3 + 4 + 5 = 15).

34. What is the purpose of the scanf() function in C?

a) To print output to the console

b) To read input from the user

c) To allocate memory dynamically

d) To define a constant

Answer: b) To read input from the user

Explanation:

- The scanf() function

Certainly, here are the remaining 6 C language MCQs:

35. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
   int x = 10;
   x >>= 2;
   printf("%d", x);
```

```
    return 0;
}
```

a) 2 b) 4 c) 5 d) 20

Answer: b) 4

Explanation:

- x >>= 2 is a bitwise right shift operator. It shifts the bits of x two positions to the right, which is equivalent to dividing x by 2^2 (4).

36. What is the purpose of the & operator in C when used with pointers?

a) To dereference a pointer

b) To get the address of a variable

c) To perform bitwise AND operation

d) To compare two pointers

Answer: b) To get the address of a variable

Explanation:

- The & operator is the address-of operator. When used with a variable, it returns the memory address of that variable.

37. What is the output of the following code?

C

```
#include <stdio.h>
int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    printf("%d", *(ptr + 2));
    return 0;
}
```

a) 1 b) 2 c) 3 d) 4

Answer: c) 3

Explanation:

- ptr points to the first element of the array.
- *(ptr + 2) accesses the third element of the array (since array indices start from 0).

38. What is the purpose of the NULL pointer in C?

a) To represent an invalid memory address

b) To represent the address of the first element of an array

c) To represent the address of the last element of an array

d) To represent a pointer to a constant

Answer: a) To represent an invalid memory address

Explanation:

- NULL is a special constant that represents an invalid memory address. It is often used to initialize pointers that do not initially point to any valid memory location.

39. What is the output of the following code?

C

```
#include <stdio.h>

int main() {
   char str[] = "Hello";
   printf("%c", str[2]);
   return 0;
}
```

a) H b) e c) l d) o

Answer: c) l

Explanation:

- str[2] accesses the third character in the string (since array indices start from 0).

40. What is the purpose of the const keyword in C?

a) To declare a constant variable

b) To declare a pointer that cannot be changed

c) To declare a function that does not modify its arguments

d) All of the above

Answer: d) All of the above

Explanation:

- The const keyword can be used to:
    - Declare a constant variable (e.g., const int PI = 3.14159;)
    - Declare a pointer that cannot be changed (e.g., const int *ptr;)

- Declare a function that does not modify its arguments (e.g., void func(const int *arr);)