Data Types:

1. Which of the following is a basic data type in C?
   - a) Array
   - b) Integer
   - c) Function
   - d) Structure

   Correct answer: b) Integer
2. What is the size of the 'int' data type in C?
   - a) 2 bytes
   - b) 4 bytes
   - c) 8 bytes
   - d) 1 byte

   Correct answer: b) 4 bytes
3. Which of the following data types can store decimal values?
   - a) int
   - b) char
   - c) float
   - d) short

   Correct answer: c) float
4. What is the range of an 'unsigned int' in C?
   - a) 0 to 32767
   - b) 0 to 65535
   - c) -32768 to 32767
   - d) 0 to 4294967295

   Correct answer: d) 0 to 4294967295
5. Which of the following data types is used to store large numbers in C?
   - a) char
   - b) double
   - c) long
   - d) float

   Correct answer: c) long
6. What is the correct size of a 'char' type in C?
   - a) 2 bytes
   - b) 1 byte
   - c) 4 bytes
   - d) 8 bytes

   Correct answer: b) 1 byte
7. What is the default value of an uninitialized 'int' variable in C?
   - a) 0
   - b) Undefined
   - c) Garbage value
   - d) NULL

   Correct answer: c) Garbage value
8. Which of the following is a non-primitive data type in C?
   - a) int
   - b) char
   - c) float
   - d) struct

   Correct answer: d) struct
9. What does the 'long long' data type represent in C?
   - a) A larger floating-point number
   - b) A larger integer number
   - c) A larger character set
   - d) A double-precision floating-point number

   Correct answer: b) A larger integer number
10. What is the size of a 'double' data type in C?
    - a) 2 bytes
    - b) 4 bytes
    - c) 8 bytes
    - d) 16 bytes

    Correct answer: c) 8 bytes
11. Which of the following types is used for storing boolean values in C?
    - a) bool
    - b) int
    - c) char
    - d) _Bool

    Correct answer: d) _Bool
12. Which of the following defines the void data type in C?
    - a) A data type with no value
    - b) A data type to store any number
    - c) A pointer to any type
    - d) A data type for characters

    Correct answer: a) A data type with no value
13. What does the 'unsigned' keyword do when applied to an integer data type?
    - a) Allows negative values
    - b) Increases the range of positive values
    - c) Reduces the size of the integer
    - d) Makes it a floating-point number

    Correct answer: b) Increases the range of positive values
14. Which type can hold a value up to 3.4e+38 in C?
    - a) int
    - b) double
    - c) char
    - d) long

    Correct answer: b) double
15. Which of the following is a valid way to declare an integer variable?
    - a) int var;
    - b) integer var;
    - c) int var = 10;
    - d) b) and c)

    Correct answer: a) int var;

---

2. Operators:

1. Which operator is used to access a member of a structure through a pointer?
   - a) .
   - b) ->
   - c) &
   - d) *

   Correct answer: b) ->
2. Which operator is used to check if two values are equal in C?
   - a) ==
   - b) =
   - c) !=
   - d) <=

   Correct answer: a) ==
3. Which operator in C is used for division?
   - a) /
   - b) *
   - c) +
   - d) %

   Correct answer: a) /
4. What does the '&&' operator represent in C?
   - a) OR operator
   - b) AND operator
   - c) NOT operator
   - d) Assigns value

   Correct answer: b) AND operator
5. Which operator is used to get the address of a variable in C?
   - a) *
   - b) &
   - c) #
   - d) $

   Correct answer: b) &
6. Which of the following operators is used to increment a value by 1?
   - a) --
   - b) ==
   - c) ++
   - d) +

   Correct answer: c) ++
7. What is the output of the expression 5 % 2?
   - a) 2
   - b) 1
   - c) 5
   - d) 0

   Correct answer: b) 1
8. What will the expression '5 / 2' return in C?

- a) 2.5
- b) 2
- c) 3
- d) 5

Correct answer: b) 2

9. Which of the following is a relational operator in C?
   - a) +
   - b) &
   - c) ==
   - d) *

   Correct answer: c) ==
10. Which operator is used to multiply two numbers in C?
    - a) /
    - b) %
    - c) *
    - d) +

    Correct answer: c) *
11. Which of the following is the correct assignment operator in C?
    - a) -
    - b) =
    - c) ==
    - d) +

    Correct answer: b) =
12. Which operator is used for logical negation in C?
    - a) !
    - b) ~
    - c) ^
    - d) &&

    Correct answer: a) !
13. What does the 'sizeof' operator do in C?
    - a) Calculates the size of the operand
    - b) Returns the address of the operand
    - c) Computes the value of the operand
    - d) None of the above

    Correct answer: a) Calculates the size of the operand
14. Which operator is used for bitwise AND in C?
    - a) |
    - b) &
    - c) ^
    - d) ~

    Correct answer: b) &
15. Which operator is used to access elements of an array in C?
    - a) .
    - b) ->
    - c) []
    - d) {}

    Correct answer: c) []

---

3. Fundamentals:

1. What is the output of the following C code?

   ```
   int main() {
       printf("%d", 10 / 3);
       return 0;
   }
   ```

   - a) 3
   - b) 3.333
   - c) 0
   - d) 10

   Correct answer: a) 3
2. Which function is used to read a character from the user in C?
   - a) get()
   - b) input()
   - c) getchar()
   - d) scanf()

   Correct answer: c) getchar()
3. How can we stop a loop in C?
   - a) break
   - b) continue
   - c) exit
   - d) return

   Correct answer: a) break
4. What is the correct syntax for a 'while' loop in C?
   - a) while condition { }
   - b) while (condition) { }
   - c) while [condition] { }
   - d) while {condition}

   Correct answer: b) while (condition) { }
5. Which of the following is used to declare a function prototype in C?
   - a) void function();
   - b) function void();
   - c) int function;
   - d) int function();

   Correct answer: a) void function();
6. Which header file is required for using the 'printf' and 'scanf' functions in C?
   - a) conio.h
   - b) math.h
   - c) stdio.h
   - d) string.h

   Correct answer: c) stdio.h
7. What will be the output of the following C code?

   ```
   int a = 5, b = 2;
   printf("%d", a / b);
   ```

   - a) 2.5
   - b) 2
   - c) 5
   - d) 0

   Correct answer: b) 2
8. Which of the following is the correct syntax for a 'for' loop in C?
   - a) for (int i = 0; i < 10; i++)
   - b) for i = 0 to 10
   - c) for (int i: 0 to 10)
   - d) None of the above

   Correct answer: a) for (int i = 0; i < 10; i++)
9. How is memory allocated dynamically in C?
   - a) malloc()
   - b) free()
   - c) alloc()
   - d) realloc()

   Correct answer: a) malloc()
10. What is the keyword used to define a constant value in C?
    - a) const
    - b) constant
    - c) #define
    - d) static

    Correct answer: a) const
11. Which of the following is the correct syntax for using a comment in C?
    - a) // comment
    - b) /* comment */
    - c) ** comment
    - d) Both a) and b)

    Correct answer: d) Both a) and b)
12. What does the 'return' statement do in C?
    - a) Exits the program
    - b) Returns a value to the calling function
    - c) Prints a value
    - d) None of the above

    Correct answer: b) Returns a value to the calling function
13. What will the following code print?

    ```
    printf("%d", 10 / 4);
    ```

    - a) 2
    - b) 2.5

- c) 0
- d) 4

  Correct answer: a) 2

14. Which of the following operators is used to check if two values are unequal?
    - a) ==
    - b) !=
    - c) <
    - d) <=

    Correct answer: b) !=
15. How are arrays declared in C?
    - a) int arr[10];
    - b) int[10] arr;
    - c) array int[10];
    - d) arr[10] int;

    Correct answer: a) int arr[10];

4. Control Structures:

1. What is the output of the following C code?

```
int x = 5;
if(x > 3) {
   printf("Hello");
}
```

   - a) 5
   - b) Hello
   - c) Nothing
   - d) Error

   Correct answer: b) Hello
2. Which of the following statements is used to exit a loop in C?
   - a) continue
   - b) break
   - c) exit
   - d) return

   Correct answer: b) break
3. What will the following C code print?

```
int x = 5;
switch(x) {
   case 5:
      printf("Yes");
      break;
   default:
      printf("No");
}
```

   - a) Yes
   - b) No
   - c) Error
   - d) Nothing

   Correct answer: a) Yes
4. How do you represent an 'else if' condition in C?
   - a) else if (condition)
   - b) else if condition
   - c) elif (condition)
   - d) else (condition)

   Correct answer: a) else if (condition)
5. What does the 'continue' statement do inside a loop in C?
   - a) Exits the loop
   - b) Skips the current iteration and continues to the next iteration
   - c) Breaks the loop completely
   - d) Restarts the loop

   Correct answer: b) Skips the current iteration and continues to the next iteration
6. Which of the following is the correct syntax for an 'if' statement in C?
   - a) if condition {}
   - b) if (condition) {}
   - c) if condition then {}
   - d) if (condition) then {}

   Correct answer: b) if (condition) {}
7. What will be printed by the following C code?

```
int i = 0;
while(i < 3) {
   printf("%d", i);
   i++;
}
```

   - a) 012
   - b) 123
   - c) 0
   - d) Error

   Correct answer: a) 012
8. What is the output of the following code?

```
int x = 4;
if(x == 4) {
   printf("True");
} else {
   printf("False");
}
```

   - a) True
   - b) False
   - c) Nothing
   - d) Error

   Correct answer: a) True
9. Which loop will execute at least once in C?
   - a) for
   - b) while
   - c) do-while
   - d) None of the above

   Correct answer: c) do-while
10. What will be the output of the following C code?

```
int x = 10;
if(x > 5)
   if(x < 20)
      printf("In range");
```

    - a) In range
    - b) 10
    - c) Error
    - d) Nothing

    Correct answer: a) In range
11. Which of the following statements is true for a switch statement in C?
    - a) It can only have 1 case
    - b) The case values must be constants
    - c) You cannot have a break statement
    - d) It works only for integer variables

    Correct answer: b) The case values must be constants
12. What does the following loop print?

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

    - a) 01234
    - b) 12345
    - c) 012
    - d) Error

    Correct answer: a) 01234
13. In a for loop, which part is executed only once?
    - a) Initialization
    - b) Condition
    - c) Increment
    - d) All parts are executed once

    Correct answer: a) Initialization

14. What is the output of the following C code?

```
int i = 10;
while(i--) {
   printf("%d", i);
}
```

   - a) 9
   - b) 10
   - c) 9876543210
   - d) Error

   Correct answer: c) 9876543210

15. What is the purpose of the 'goto' statement in C?
   - a) It is used to jump to a specific point in the program
   - b) It is used to define a function
   - c) It is used to declare variables
   - d) It is used to declare constants

   Correct answer: a) It is used to jump to a specific point in the program

---

5. Arrays:

1. What is the correct way to declare an array of 10 integers in C?
   - a) int arr[10];
   - b) int[10] arr;
   - c) arr(10) int;
   - d) None of the above

   Correct answer: a) int arr[10];

2. How do you access the third element of an array named 'arr' in C?
   - a) arr[3]
   - b) arr[2]
   - c) arr{2}
   - d) arr[1]

   Correct answer: b) arr[2]

3. What will the following code print?

```
int arr[] = {1, 2, 3, 4};
printf("%d", arr[2]);
```

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

   Correct answer: c) 3

4. Which of the following is the correct way to initialize an array in C?
   - a) int arr[] = {1, 2, 3};
   - b) int arr(3) = {1, 2, 3};
   - c) int arr[3] = {1, 2, 3};
   - d) All of the above

   Correct answer: a) int arr[] = {1, 2, 3};

5. What is the output of the following code?

```
int arr[] = {10, 20, 30};
printf("%d", arr[1]);
```

   - a) 10
   - b) 20
   - c) 30
   - d) Error

   Correct answer: b) 20

6. How do you declare an array of 5 floats in C?
   - a) float arr[5];
   - b) float[5] arr;
   - c) arr[5] float;
   - d) None of the above

   Correct answer: a) float arr[5];

7. What is the correct index of the last element in an array of size 5?
   - a) 5
   - b) 4
   - c) 3
   - d) 6

   Correct answer: b) 4

8. Which function is used to determine the length of a string in C?
   - a) length()
   - b) size()
   - c) strlen()
   - d) sizeof()

   Correct answer: c) strlen()

9. How do you initialize an array of 5 integers to all zeros?
   - a) int arr[5] = {0};
   - b) int arr[5] = {};
   - c) int arr[5] = {0, 0, 0, 0, 0};
   - d) None of the above

   Correct answer: a) int arr[5] = {0};

10. What will the following code print?

```
int arr[5] = {1, 2, 3, 4, 5};
printf("%d", arr[3]);
```

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

   Correct answer: d) 4

11. What is the size of the array int arr[10] in bytes?
   - a) 10 bytes
   - b) 20 bytes
   - c) 40 bytes
   - d) 50 bytes

   Correct answer: b) 20 bytes

12. Which of the following can be used to change the size of an array dynamically?
   - a) realloc()
   - b) malloc()
   - c) sizeof()
   - d) None of the above

   Correct answer: a) realloc()

13. How do you pass an array to a function in C?
   - a) By value
   - b) By reference
   - c) By copying
   - d) By pointer

   Correct answer: b) By reference

14. What does the following code do?

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

   - a) It prints the array
   - b) It changes the value of the second element to 5
   - c) It deletes the second element
   - d) Nothing

   Correct answer: b) It changes the value of the second element to 5

15. What will be the output of this code?

```
int arr[3] = {1, 2, 3};
printf("%d", arr[2]);
```

   - a) 1
   - b) 2
   - c) 3
   - d) Error

   Correct answer: c) 3

6. String Handling Functions:

1. Which of the following is used to find the length of a string in C?
    - a) length()
    - b) size()
    - c) strlen()
    - d) strlength()

    Correct answer: c) strlen()
2. What is the function used to concatenate two strings in C?
    - a) strcat()
    - b) strcpy()
    - c) strconcat()
    - d) append()

    Correct answer: a) strcat()
3. Which function is used to copy a string from one variable to another in C?
    - a) strcpy()
    - b) strncpy()
    - c) copystr()
    - d) strcopy()

    Correct answer: a) strcpy()
4. What does the function strcmp() do in C?
    - a) Compares two strings
    - b) Copies one string to another
    - c) Finds the length of a string
    - d) Concatenates two strings

    Correct answer: a) Compares two strings
5. What is the return type of the strcpy() function in C?
    - a) int
    - b) char
    - c) char *
    - d) void

    Correct answer: c) char *
6. What will be the output of the following code?

    ```
    char str[] = "Hello";
    printf("%c", str[1]);
    ```

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

    Correct answer: b) e
7. What is the correct way to declare a string in C?
    - a) char str = "Hello";
    - b) char str[] = "Hello";
    - c) string str = "Hello";
    - d) char str[5] = "Hello";

    Correct answer: b) char str[] = "Hello";
8. Which function is used to convert a string to an integer in C?
    - a) atoi()
    - b) itoa()
    - c) atoi_str()
    - d) str_to_int()

    Correct answer: a) atoi()
9. What does the function strrev() do in C?
    - a) Reverses a string
    - b) Returns the reverse of a string
    - c) Converts a string to uppercase
    - d) Concatenates a string

    Correct answer: a) Reverses a string
10. How do you find the first occurrence of a character in a string in C?
    - a) strchr()
    - b) strchrn()
    - c) findchar()
    - d) strstr()

    Correct answer: a) strchr()
11. Which function returns the position of a substring in a string in C?
    - a) strstr()
    - b) substring()
    - c) strfind()
    - d) strloc()

    Correct answer: a) strstr()
12. Which function returns the length of a string in C?
    - a) length()
    - b) strlength()
    - c) strlen()
    - d) sizeof()

    Correct answer: c) strlen()
13. What does the function strtok() do in C?
    - a) Tokenizes a string
    - b) Compares two strings
    - c) Concatenates two strings
    - d) Copies one string to another

    Correct answer: a) Tokenizes a string
14. What will the following code print?

    ```
    char str[] = "C Programming";
    printf("%s", str);
    ```

    - a) C
    - b) Programming
    - c) C Programming
    - d) Error

    Correct answer: c) C Programming
15. What is the function used to convert a string to uppercase in C?
    - a) strupr()
    - b) strtoupper()
    - c) toupper()
    - d) upper()

    Correct answer: a) strupr()

---

7. User-Defined Functions:

1. How do you declare a function in C?
    - a) void function_name();
    - b) function_name();
    - c) function_name(void);
    - d) void function_name(void);

    Correct answer: d) void function_name(void);
2. What is the correct syntax for a function that returns an integer and takes no arguments in C?
    - a) int function() {}
    - b) function int() {}
    - c) int function(void) {}
    - d) void function(int) {}

    Correct answer: a) int function() {}
3. What will be the output of the following code?

    ```
    int sum(int a, int b) {
       return a + b;
    }
    printf("%d", sum(5, 10));
    ```

    - a) 10
    - b) 15
    - c) 50
    - d) Error

    Correct answer: b) 15
4. Which of the following is a correct function definition in C?
    - a) void add() {}
    - b) add() {}
    - c) function void add() {}
    - d) void add(int, int) {}

    Correct answer: d) void add(int, int) {}
5. What is the purpose of the return statement in a function in C?
    - a) To exit the function
    - b) To pass a value back to the calling function
    - c) To stop the execution of the program
    - d) None of the above

    Correct answer: b) To pass a value back to the calling function

6. Which function is called automatically when the program starts in C?
    - a) start()
    - b) main()
    - c) init()
    - d) setup()

    Correct answer: b) main()
7. What is the default return type of a function if no return type is specified in C?
    - a) int
    - b) void
    - c) char
    - d) float

    Correct answer: a) int
8. What will be the output of the following code?

```
int add(int a, int b) {
   return a + b;
}
printf("%d", add(5, 3));
```

- a) 5
- b) 3
- c) 8
- d) Error

Correct answer: c) 8

9. Which keyword is used to define a function in C?
    - a) func
    - b) define
    - c) void
    - d) return

    Correct answer: c) void
10. What happens if you do not provide a return statement in a non-void function in C?
    - a) It causes a syntax error
    - b) It returns a garbage value
    - c) It returns 0 by default
    - d) It exits the program

    Correct answer: b) It returns a garbage value
11. Can a function call itself in C?
    - a) Yes, it's called recursion
    - b) No, it causes an error
    - c) Yes, but only if it's void
    - d) Yes, but only with a return type

    Correct answer: a) Yes, it's called recursion
12. Which of the following is true about functions in C?
    - a) A function can call itself recursively
    - b) A function cannot return a value
    - c) A function must have a parameter
    - d) None of the above

    Correct answer: a) A function can call itself recursively
13. What will be the output of the following code?

```
int multiply(int a, int b) {
   return a * b;
}
printf("%d", multiply(4, 5));
```

- a) 9
- b) 20
- c) 45
- d) Error

Correct answer: b) 20

14. What is the correct syntax for calling a function in C?
    - a) call function_name();
    - b) function_name();
    - c) call(function_name);
    - d) None of the above

    Correct answer: b) function_name();
15. How do you pass an argument to a function in C?
    - a) By value
    - b) By reference
    - c) By pointer
    - d) All of the above

    Correct answer: d) All of the above

6. String Handling Functions:

1. Which of the following is used to find the length of a string in C?
    - a) length()
    - b) size()
    - c) strlen()
    - d) strlength()

    Correct answer: c) strlen()
2. What is the function used to concatenate two strings in C?
    - a) strcat()
    - b) strcpy()
    - c) strconcat()
    - d) append()

    Correct answer: a) strcat()
3. Which function is used to copy a string from one variable to another in C?
    - a) strcpy()
    - b) strncpy()
    - c) copystr()
    - d) strcopy()

    Correct answer: a) strcpy()
4. What does the function strcmp() do in C?
    - a) Compares two strings
    - b) Copies one string to another
    - c) Finds the length of a string
    - d) Concatenates two strings

    Correct answer: a) Compares two strings
5. What is the return type of the strcpy() function in C?
    - a) int
    - b) char
    - c) char *
    - d) void

    Correct answer: c) char *
6. What will be the output of the following code?

```
char str[] = "Hello";
printf("%c", str[1]);
```

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

Correct answer: b) e

7. What is the correct way to declare a string in C?
    - a) char str = "Hello";
    - b) char str[] = "Hello";
    - c) string str = "Hello";
    - d) char str[5] = "Hello";

    Correct answer: b) char str[] = "Hello";
8. Which function is used to convert a string to an integer in C?
    - a) atoi()
    - b) itoa()
    - c) atoi_str()
    - d) str_to_int()

    Correct answer: a) atoi()
9. What does the function strrev() do in C?
    - a) Reverses a string
    - b) Returns the reverse of a string
    - c) Converts a string to uppercase
    - d) Concatenates a string

    Correct answer: a) Reverses a string
10. How do you find the first occurrence of a character in a string in C?
    - a) strchr()
    - b) strchrn()
    - c) findchar()
    - d) strstr()

    Correct answer: a) strchr()

11. Which function returns the position of a substring in a string in C?
    - a) strstr()
    - b) substring()
    - c) strfind()
    - d) strloc()

    Correct answer: a) strstr()
12. Which function returns the length of a string in C?
    - a) length()
    - b) strlength()
    - c) strlen()
    - d) sizeof()

    Correct answer: c) strlen()
13. What does the function strtok() do in C?
    - a) Tokenizes a string
    - b) Compares two strings
    - c) Concatenates two strings
    - d) Copies one string to another

    Correct answer: a) Tokenizes a string
14. What will the following code print?

    ```
    char str[] = "C Programming";
    printf("%s", str);
    ```

    - a) C
    - b) Programming
    - c) C Programming
    - d) Error

    Correct answer: c) C Programming
15. What is the function used to convert a string to uppercase in C?
    - a) strupr()
    - b) strtoupper()
    - c) toupper()
    - d) upper()

    Correct answer: a) strupr()

---

7. User-Defined Functions:

1. How do you declare a function in C?
    - a) void function_name();
    - b) function_name();
    - c) function_name(void);
    - d) void function_name(void);

    Correct answer: d) void function_name(void);
2. What is the correct syntax for a function that returns an integer and takes no arguments in C?
    - a) int function() {}
    - b) function int() {}
    - c) int function(void) {}
    - d) void function(int) {}

    Correct answer: a) int function() {}
3. What will be the output of the following code?

    ```
    int sum(int a, int b) {
        return a + b;
    }
    printf("%d", sum(5, 10));
    ```

    - a) 10
    - b) 15
    - c) 50
    - d) Error

    Correct answer: b) 15
4. Which of the following is a correct function definition in C?
    - a) void add() {}
    - b) add() {}
    - c) function void add() {}
    - d) void add(int, int) {}

    Correct answer: d) void add(int, int) {}
5. What is the purpose of the return statement in a function in C?
    - a) To exit the function
    - b) To pass a value back to the calling function
    - c) To stop the execution of the program
    - d) None of the above

    Correct answer: b) To pass a value back to the calling function
6. Which function is called automatically when the program starts in C?
    - a) start()
    - b) main()
    - c) init()
    - d) setup()

    Correct answer: b) main()
7. What is the default return type of a function if no return type is specified in C?
    - a) int
    - b) void
    - c) char
    - d) float

    Correct answer: a) int
8. What will be the output of the following code?

    ```
    int add(int a, int b) {
        return a + b;
    }
    printf("%d", add(5, 3));
    ```

    - a) 5
    - b) 3
    - c) 8
    - d) Error

    Correct answer: c) 8
9. Which keyword is used to define a function in C?
    - a) func
    - b) define
    - c) void
    - d) return

    Correct answer: c) void
10. What happens if you do not provide a return statement in a non-void function in C?
    - a) It causes a syntax error
    - b) It returns a garbage value
    - c) It returns 0 by default
    - d) It exits the program

    Correct answer: b) It returns a garbage value
11. Can a function call itself in C?
    - a) Yes, it's called recursion
    - b) No, it causes an error
    - c) Yes, but only if it's void
    - d) Yes, but only with a return type

    Correct answer: a) Yes, it's called recursion
12. Which of the following is true about functions in C?
    - a) A function can call itself recursively
    - b) A function cannot return a value
    - c) A function must have a parameter
    - d) None of the above

    Correct answer: a) A function can call itself recursively
13. What will be the output of the following code?

    ```
    int multiply(int a, int b) {
        return a * b;
    }
    printf("%d", multiply(4, 5));
    ```

    - a) 9
    - b) 20
    - c) 45
    - d) Error

    Correct answer: b) 20
14. What is the correct syntax for calling a function in C?
    - a) call function_name();
    - b) function_name();

- c) call(function_name);
- d) None of the above

  Correct answer: b) function_name();

15. How do you pass an argument to a function in C?
    - a) By value
    - b) By reference
    - c) By pointer
    - d) All of the above

      Correct answer: d) All of the above

Call by Value and Call by Reference:

1. What is the main difference between call by value and call by reference in C?
    - a) In call by value, actual parameters are passed; in call by reference, only addresses are passed.
    - b) In call by value, addresses are passed; in call by reference, actual parameters are passed.
    - c) In call by value, copies of the actual parameters are passed; in call by reference, copies of the formal parameters are passed.
    - d) No difference between the two.

      Correct answer: a) In call by value, actual parameters are passed; in call by reference, only addresses are passed.
2. Which of the following allows the called function to modify the actual argument in C?
    - a) Call by value
    - b) Call by reference
    - c) Call by pointer
    - d) None of the above

      Correct answer: b) Call by reference
3. What happens when arguments are passed by value in C?
    - a) The function can modify the values of the actual parameters.
    - b) The function receives a copy of the values and cannot modify the original data.
    - c) The values are passed by reference.
    - d) None of the above

      Correct answer: b) The function receives a copy of the values and cannot modify the original data.
4. Which of the following is an advantage of call by reference over call by value?
    - a) Call by reference is faster because no memory is used for copying the arguments.
    - b) Call by reference is easier to implement.
    - c) Call by reference is always safer than call by value.
    - d) None of the above

      Correct answer: a) Call by reference is faster because no memory is used for copying the arguments.
5. Which of the following is used to pass arguments by reference in C?
    - a) Using the ampersand (&) symbol
    - b) Using the asterisk (*) symbol
    - c) By passing the variable directly
    - d) By using pointers

      Correct answer: d) By using pointers
6. What will the following code output?

    ```
    void change(int a) {
        a = 10;
    }
    int main() {
        int num = 5;
        change(num);
        printf("%d", num);
        return 0;
    }
    ```

    - a) 5
    - b) 10
    - c) Error
    - d) Undefined behavior

      Correct answer: a) 5
7. What will the following code output?

    ```
    void change(int *a) {
        *a = 10;
    }
    int main() {
        int num = 5;
        change(&num);
        printf("%d", num);
        return 0;
    }
    ```

    - a) 5
    - b) 10
    - c) Error
    - d) Undefined behavior

      Correct answer: b) 10
8. Which function call method is typically used in recursion in C?
    - a) Call by value
    - b) Call by reference
    - c) Call by pointer
    - d) None of the above

      Correct answer: a) Call by value
9. In which of the following methods can we pass large data without affecting performance?
    - a) Call by value
    - b) Call by reference
    - c) Both a and b
    - d) None of the above

      Correct answer: b) Call by reference
10. Which of the following is true about passing arrays to functions in C?
    - a) Arrays are always passed by value.
    - b) Arrays are passed by reference.
    - c) Arrays cannot be passed to functions in C.
    - d) None of the above

      Correct answer: b) Arrays are passed by reference.
11. What does the following code snippet do in C?

    ```
    void increment(int *x) {
        (*x)++;
    }
    ```

    - a) Increments the value of x
    - b) Increments the address of x
    - c) Increments the memory location of x
    - d) Increments the pointer x itself

      Correct answer: a) Increments the value of x
12. What will be the output of the following code?

    ```
    void change(int a) {
        a = 10;
    }
    int main() {
        int num = 5;
        change(num);
        printf("%d", num);
    }
    ```

    - a) 5
    - b) 10
    - c) Error
    - d) Undefined behavior

      Correct answer: a) 5
13. Which of the following is a disadvantage of call by reference?
    - a) It allows direct modification of actual arguments.
    - b) It can cause unintentional modification of the arguments.
    - c) It is slower than call by value.

- d) None of the above

  Correct answer: b) It can cause unintentional modification of the arguments.

14. Which of the following best describes call by reference?
    - a) Passing a copy of the actual value.
    - b) Passing the memory address of the argument.
    - c) Passing the value to the pointer.
    - d) Passing a reference to a pointer.

    Correct answer: b) Passing the memory address of the argument.
15. In C, if a function modifies the argument, but the changes are not reflected outside the function, which call method is used?
    - a) Call by value
    - b) Call by reference
    - c) Call by pointer
    - d) None of the above

    Correct answer: a) Call by value

---

9. Storage Classes:

1. Which storage class defines variables that retain their values between function calls in C?
   - a) auto
   - b) extern
   - c) static
   - d) register

   Correct answer: c) static
2. What is the default storage class of a local variable in C?
   - a) static
   - b) auto
   - c) register
   - d) extern

   Correct answer: b) auto
3. Which of the following storage classes makes a variable accessible across multiple files in C?
   - a) static
   - b) auto
   - c) extern
   - d) register

   Correct answer: c) extern
4. What will happen if a variable is declared as register in C?
   - a) It will store the value in a register of the CPU, if available.
   - b) It will be allocated in the heap.
   - c) It will be stored in the stack.
   - d) None of the above

   Correct answer: a) It will store the value in a register of the CPU, if available.
5. In which of the following scenarios is a static variable useful in C?
   - a) When a variable needs to be initialized only once and retains its value between function calls.
   - b) When a variable is shared between different functions.
   - c) When a variable should not be initialized.
   - d) When a variable needs to be stored in the heap.

   Correct answer: a) When a variable needs to be initialized only once and retains its value between function calls.
6. What is the lifetime of a variable with static storage class in C?
   - a) It exists until the program terminates.
   - b) It exists for the duration of the function call.
   - c) It exists only for the scope of the function.
   - d) It exists for the duration of the program execution.

   Correct answer: a) It exists until the program terminates.
7. What is the storage class of the following variable?

   ```
   static int a;
   ```

   - a) auto
   - b) static
   - c) extern
   - d) register

   Correct answer: b) static
8. What is the main purpose of the extern keyword in C?
   - a) To define a variable with global scope
   - b) To make a variable visible across different files
   - c) To restrict a variable's scope within a function
   - d) To allocate memory dynamically

   Correct answer: b) To make a variable visible across different files
9. Can a variable with extern storage class be initialized in C?
   - a) Yes
   - b) No
   - c) It depends on the compiler
   - d) Only within the main function

   Correct answer: b) No
10. Which storage class is used to give a variable a scope that is limited to the current block in C?
    - a) auto
    - b) static
    - c) register
    - d) extern

    Correct answer: a) auto
11. What will be the value of a static variable if it is not initialized explicitly in C?
    - a) Random value
    - b) 0
    - c) Undefined
    - d) Compiler dependent

    Correct answer: b) 0
12. What is the primary purpose of the register storage class in C?
    - a) To store the variable in the register
    - b) To make the variable available globally
    - c) To make the variable persist between function calls
    - d) To store the variable in the heap

    Correct answer: a) To store the variable in the register
13. Can register variables be used with the & operator in C?
    - a) Yes
    - b) No
    - c) Only in certain cases
    - d) Depends on the compiler

    Correct answer: b) No
14. What is the default storage class for global variables in C?
    - a) auto
    - b) static
    - c) extern
    - d) register

    Correct answer: c) extern
15. What happens when an extern variable is declared in a C file?
    - a) It is allocated memory in the file.
    - b) It refers to a variable that is defined in another file.
    - c) It cannot be used in the file.
    - d) It is treated as a constant.

    Correct answer: b) It refers to a variable that is defined in another file.

Recursion:

1. What is recursion in C?
   - a) A function that calls itself
   - b) A function that calls another function
   - c) A function that returns a value
   - d) A function that is always iterative

   Correct answer: a) A function that calls itself
2. In which of the following scenarios can recursion be useful in C?
   - a) To solve problems that can be divided into smaller sub-problems
   - b) To reduce memory usage
   - c) To make the code more complex

- d) None of the above

  Correct answer: a) To solve problems that can be divided into smaller sub-problems

3. What is the base case in a recursive function?
   - a) A condition that causes the recursion to terminate
   - b) A condition that causes the function to call itself
   - c) A condition that causes the program to exit
   - d) A condition that causes an infinite loop

   Correct answer: a) A condition that causes the recursion to terminate
4. What will happen if a recursive function does not have a proper base case in C?
   - a) The function will never terminate and cause a stack overflow
   - b) The function will terminate after some iterations
   - c) The program will output an error message
   - d) The program will skip the function

   Correct answer: a) The function will never terminate and cause a stack overflow
5. Consider the following recursive function:

   ```
   int factorial(int n) {
      if (n == 0) {
         return 1;
      }
      return n * factorial(n-1);
   }
   ```

   What does this function compute?

   - a) The sum of numbers from 1 to n
   - b) The factorial of a number n
   - c) The Fibonacci number at position n
   - d) None of the above

   Correct answer: b) The factorial of a number n
6. What is the time complexity of the recursive factorial function?
   - a) O(1)
   - b) O(log n)
   - c) O(n)
   - d) O(n^2)

   Correct answer: c) O(n)
7. Which of the following is an example of a problem that can be solved using recursion?
   - a) Calculating the greatest common divisor (GCD)
   - b) Printing Fibonacci numbers
   - c) Sorting an array
   - d) All of the above

   Correct answer: d) All of the above
8. Which of the following is true for recursive functions in C?
   - a) They always perform better than iterative solutions.
   - b) They can lead to a stack overflow if not used carefully.
   - c) They require fewer lines of code than iterative solutions.
   - d) They do not need a base case to work.

   Correct answer: b) They can lead to a stack overflow if not used carefully.
9. What happens in the following recursive function call?

   ```
   void func(int n) {
      if (n == 0) return;
      printf("%d ", n);
      func(n-1);
   }
   ```

   - a) It prints numbers from n down to 0
   - b) It prints numbers from 0 to n
   - c) It causes an infinite loop
   - d) It prints nothing

   Correct answer: a) It prints numbers from n down to 0
10. What is the output of the following recursive function?

    ```
    void func(int n) {
       if (n > 0) {
          func(n - 1);
          printf("%d ", n);
       }
    }
    int main() {
       func(3);
       return 0;
    }
    ```

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

    Correct answer: b) 1 2 3
11. What is the output of the following code?

    ```
    int fibonacci(int n) {
       if (n <= 1) return n;
       return fibonacci(n - 1) + fibonacci(n - 2);
    }
    int main() {
       printf("%d", fibonacci(5));
       return 0;
    }
    ```

    - a) 5
    - b) 8
    - c) 3
    - d) 13

    Correct answer: a) 5
12. What is the advantage of using recursion in a problem?
    - a) Recursion simplifies the code and logic for complex problems
    - b) Recursion is faster than iteration
    - c) Recursion always uses less memory
    - d) Recursion is always more efficient

    Correct answer: a) Recursion simplifies the code and logic for complex problems
13. Which of the following problems is commonly solved using recursion in C?
    - a) Tower of Hanoi
    - b) Sorting an array
    - c) Searching in a sorted list
    - d) All of the above

    Correct answer: a) Tower of Hanoi
14. What will happen if the function below is called with factorial(5) in C?

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

    - a) It will cause a stack overflow
    - b) It will return 120
    - c) It will return 0
    - d) It will return undefined behavior

    Correct answer: b) It will return 120

---

11. Structure and Nested Structure:

1. What is a structure in C?
   - a) A type of array
   - b) A collection of variables of different types
   - c) A collection of variables of the same type
   - d) A function

   Correct answer: b) A collection of variables of different types

2. How do you define a structure in C?
    - a) struct name { type1 var1; type2 var2; };
    - b) struct { type1 var1; type2 var2; } name;
    - c) struct name { type1 var1; };
    - d) All of the above

    Correct answer: d) All of the above
3. Can you have a structure inside another structure in C?
    - a) Yes
    - b) No
    - c) Only in specific compilers
    - d) Only with pointers

    Correct answer: a) Yes
4. What is the correct way to access members of a structure in C?
    - a) structure.member
    - b) structure->member
    - c) Both of the above
    - d) None of the above

    Correct answer: c) Both of the above
5. Which of the following is an example of a nested structure in C?
    - a) A structure that contains an array
    - b) A structure that contains another structure as a member
    - c) A structure with a pointer to another structure
    - d) None of the above

    Correct answer: b) A structure that contains another structure as a member
6. What is the output of the following C code?

```
struct person {
   char name[50];
   int age;
};
struct student {
   struct person p;
   int grade;
};
struct student s1;
s1.p.age = 20;
printf("%d", s1.p.age);
```

- a) Error
- b) 20
- c) 50
- d) Undefined

Correct answer: b) 20

7. What does the sizeof() operator return when used with a structure in C?
    - a) The number of members in the structure
    - b) The size of the structure in bytes
    - c) The size of the largest member in the structure
    - d) None of the above

    Correct answer: b) The size of the structure in bytes
8. Which of the following is the correct syntax to define a structure pointer in C?
    - a) struct name *ptr;
    - b) struct *ptr;
    - c) name *ptr;
    - d) struct *ptr[];

    *_Correct answer: a) struct name ptr;_
9. What is the output of the following code?

```
struct complex {
   int real;
   int imag;
};
struct complex num;
num.real = 3;
num.imag = 4;
printf("%d + %di", num.real, num.imag);
```

- a) 3 + 4i
- b) 3 + 4
- c) 7
- d) Error

Correct answer: a) 3 + 4i

10. Which of the following is true for a structure in C?
    - a) Structure members can only be integers
    - b) Structure members can have different data types
    - c) Structure variables must be integers
    - d) Structure members are always fixed-size

    Correct answer: b) Structure members can have different data types

Union:

1. What is the size of a union in C?
    - a) The size of the largest member of the union
    - b) The sum of the sizes of all the members
    - c) The number of members in the union
    - d) Undefined

    Correct answer: a) The size of the largest member of the union
2. Which of the following is a valid union definition in C?
    - a) union name { int a; char b; };
    - b) union { int a; float b; } name;
    - c) union name { int a; char b; } name;
    - d) All of the above

    Correct answer: d) All of the above
3. In a union, what happens if we assign a value to one member?
    - a) It overwrites the values of all other members
    - b) It does not affect other members
    - c) It stores the value for the assigned member only
    - d) It raises a compilation error

    Correct answer: a) It overwrites the values of all other members
4. What will be the output of the following code?

```
union u {
   int a;
   char b;
};
union u var;
var.a = 10;
printf("%d", var.b);
```

- a) 10
- b) 'a'
- c) Garbage value
- d) Error

Correct answer: c) Garbage value

5. What is the main difference between a structure and a union in C?
    - a) A structure allocates memory for all its members, while a union shares memory for its members
    - b) A structure allows only one member to be accessed at a time
    - c) A union allows multiple members to be accessed at once
    - d) A structure is faster to access than a union

    Correct answer: a) A structure allocates memory for all its members, while a union shares memory for its members
6. How do you access a union member in C?
    - a) Using the dot operator (union.member)
    - b) Using the arrow operator (union->member)
    - c) Using the sizeof() operator
    - d) Using both dot and arrow operators

    Correct answer: a) Using the dot operator (union.member)
7. What is the output of the following code?

```
union test {
   int i;
   float f;
};
union test t;
t.i = 5;
```

```
t.f = 10.5;
printf("%d %f", t.i, t.f);
```

- a) 5 10.5
- b) 5 0.0
- c) 0 10.5
- d) Undefined
  Correct answer: b) 5 0.0

8. Which of the following statements is true regarding unions in C?
   - a) All members of a union share the same memory location
   - b) Unions are always faster than structures
   - c) You can assign values to all members of a union simultaneously
   - d) Unions always use less memory than structures
     Correct answer: a) All members of a union share the same memory location
9. What is the maximum number of members that can be assigned values simultaneously in a union in C?
   - a) One
   - b) Two
   - c) All members
   - d) None of the above
     Correct answer: a) One
10. How do you define a union that contains an integer and a float in C?
    - a) union { int x; float y; };
    - b) union data { int x; float y; };
    - c) union int { int x; float y; };
    - d) struct { int x; float y; };
      Correct answer: b) union data { int x; float y; };

---

13. Pointers:

1. What does a pointer in C store?
   - a) The value of a variable
   - b) The address of a variable
   - c) The data type of a variable
   - d) None of the above
     Correct answer: b) The address of a variable
2. How do you declare a pointer in C?
   - a) int ptr;
   - b) int *ptr;
   - c) int ptr*;
   - d) *int ptr;
     *_Correct answer: b) int ptr;_
3. What is the output of the following C code?

```
int a = 10;
int *ptr = &a;
printf("%d", *ptr);
```

   - a) 10
   - b) &a
   - c) ptr
   - d) Undefined
     Correct answer: a) 10
4. What is the purpose of the & operator in C?
   - a) To dereference a pointer
   - b) To find the address of a variable
   - c) To declare a pointer
   - d) To compare values
     Correct answer: b) To find the address of a variable
5. Which of the following is the correct syntax to get the value pointed to by a pointer?
   - a) ptr->value
   - b) *ptr
   - c) &ptr
   - d) ptr.value
     *_Correct answer: b) ptr_
6. What is the output of the following code?

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

   - a) 5
   - b) 10
   - c) Undefined
   - d) Error
     Correct answer: b) 10
7. Which of the following correctly defines a pointer to a pointer in C?
   - a) int **ptr;
   - b) int *ptr*;
   - c) int ptr**;
   - d) **int ptr;
     Correct answer: a) int **ptr;
8. What is the result of dereferencing a NULL pointer in C?
   - a) It will print NULL
   - b) It will result in undefined behavior
   - c) It will cause a segmentation fault
   - d) It will cause an infinite loop
     Correct answer: b) It will result in undefined behavior
9. What is the correct way to initialize a pointer to NULL in C?
   - a) int *ptr = 0;
   - b) int *ptr = NULL;
   - c) Both a and b
   - d) None of the above
     Correct answer: c) Both a and b
10. Which of the following can be stored in a pointer?
    - a) Memory addresses
    - b) Data values
    - c) Arrays
    - d) Functions
      Correct answer: a) Memory addresses

---

14. File Handling:

1. Which of the following functions is used to open a file in C?
   - a) fopen()
   - b) fileopen()
   - c) open()
   - d) file_read()
     Correct answer: a) fopen()
2. Which of the following is the correct syntax for opening a file in write mode in C?
   - a) fopen("filename", "w");
   - b) fopen("filename", "r");
   - c) fopen("filename", "a");
   - d) fopen("filename", "rw");
     Correct answer: a) fopen("filename", "w");
3. How do you close a file in C?
   - a) fileclose()
   - b) close()
   - c) fclose()
   - d) file_end()
     Correct answer: c) fclose()
4. What will happen if you try to open a non-existent file in read mode using fopen() in C?
   - a) The program will crash
   - b) The program will display an error message
   - c) It will return NULL
   - d) It will create the file automatically
     Correct answer: c) It will return NULL
5. Which of the following file modes in C allows both reading and writing?

- a) r
- b) w
- c) a
- d) r+

Correct answer: d) r+

6. What does the fgetc() function do in C?
    - a) Reads an entire line of text
    - b) Reads a single character from a file
    - c) Reads the entire contents of a file
    - d) Writes a single character to a file

    Correct answer: b) Reads a single character from a file
7. What is the return type of the fopen() function in C?
    - a) int
    - b) FILE*
    - c) char*
    - d) void

    Correct answer: b) FILE*
8. Which function is used to write data to a file in C?
    - a) fwrite()
    - b) filewrite()
    - c) putchar()
    - d) fprintf()

    Correct answer: a) fwrite()
9. What happens when you open a file in append mode ("a") in C?
    - a) The file is overwritten
    - b) The file is read-only
    - c) The new data is written at the end of the file
    - d) The file is created if it doesn't exist

    Correct answer: c) The new data is written at the end of the file
10. What is the correct function to check if the end of a file is reached in C?
    - a) feof()
    - b) endfile()
    - c) fileend()
    - d) EOF()

    Correct answer: a) feof()