C Basics

1. What will happen if you compile and run the following code?

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

   a) 2.5
   b) 2
   c) Compiler error
   d) Undefined behavior

2. Which of the following statements about macros is correct? a) Macros are processed by the preprocessor.
   b) Macros can accept arguments.
   c) Macros are not type-checked.
   d) All of the above
3. Which of the following is valid in C? a) `int x = 1, y = x + 1;`
   b) `int x = y + 1, y = 2;`
   c) `int x, x = 5;`
   d) `int 123x = 5;`
4. What is the output of the following code?

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

   a) Compiler error
   b) Undefined behavior
   c) 5 6
   d) 6 6

5. Which of the following is a valid statement in C? a) `const int x = 5; x++;`
   b) `enum {A, B = 3, C};`
   c) `int *ptr; ptr = 10;`
   d) `float x = "abc";`

Functions

6. Which of the following is true about inline functions in C? a) Inline functions can reduce function call overhead.
   b) Inline functions must be defined in a header file.
   c) Inline functions are just a request to the compiler.
   d) All of the above
7. What will the following code output?

   ```
   #include <stdio.h>
   void func(int x) {
       x += 5;
   }
   int main() {
       int y = 10;
       func(y);
       printf("%d", y);
       return 0;
   }
   ```

   a) 10
   b) 15
   c) Undefined
   d) Compiler error

8. What is the output of this recursive function?

   ```
   #include <stdio.h>
   int mystery(int n) {
       if (n <= 0) return 1;
       return n * mystery(n - 2);
   }
   int main() {
       printf("%d", mystery(5));
       return 0;
   }
   ```

   a) 120
   b) 15
   c) 75
   d) 1

9. Can a function return multiple values in C? a) No
   b) Yes, using pointers or arrays.
   c) Yes, using multiple `return` statements.
   d) Yes, but only integers.

10. Which of the following function declarations is invalid? a) `void myFunc();`
    b) `int myFunc(int x, int y = 10);`
    c) `float myFunc(float x);`
    d) `int myFunc(...);`

Arrays

11. Predict the output of this code:

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

    a) 0
    b) 3
    c) Garbage value
    d) Compiler error

12. What will the following code output?

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

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

13. How do you pass a 2D array to a function in C? a) `void func(int arr[][]);`
    b) `void func(int *arr);`
    c) `void func(int arr[5][5]);`
    d) `void func(int **arr);`
14. What does the following code output?

```
#include <stdio.h>
int main() {
    char str[] = "Hello";
    printf("%c", str[5]);
    return 0;
}
```

    a) `o`
    b) `\0`
    c) Garbage value
    d) Compiler error

15. Which of the following is true about multidimensional arrays in C? a) They must be square.
    b) They are stored in row-major order.
    c) Only 2D arrays are allowed.
    d) Their size must be specified during runtime.

Pointers and DMA

16. What does the following code print?

```
#include <stdio.h>
int main() {
    int x = 10;
    int *ptr = &x;
    printf("%p", ptr + 1);
    return 0;
}
```

    a) Address of `x + 1`
    b) Address of `x + sizeof(int)`
    c) Compiler error
    d) Undefined behavior

17. What happens if `free()` is called twice on the same pointer? a) Memory leak
    b) Double free error
    c) Program crash
    d) Undefined behavior
18. What will the following code output?

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int *ptr = malloc(0);
    if (ptr) printf("Allocated");
    else printf("Not allocated");
    free(ptr);
    return 0;
}
```

    a) `Allocated`
    b) `Not allocated`
    c) `Undefined`
    d) Compiler error

19. What does this code output?

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

    a) 10
    b) 15
    c) Compiler error
    d) Undefined behavior

20. Which of the following correctly allocates a 2D array dynamically? a) `int *arr = malloc(rows * cols * sizeof(int));`
    b) `int **arr = malloc(rows * sizeof(int*));`
    c) `int **arr = malloc(sizeof(int) * cols);`
    d) `int arr[rows][cols];`

1. Structures

1. Which keyword is used to define a structure in C?
   a) struct
   b) structure
   c) class
   d) union
2. What is the default access modifier for members of a structure in C?
   a) private
   b) protected
   c) public
   d) None of the above
3. How can you initialize a structure in C?
   a) Using curly braces
   b) Using parentheses
   c) Using brackets
   d) None of the above
4. Can a structure contain a pointer to itself?
   a) Yes
   b) No
5. How is memory allocated for a structure in C?
   a) Sequentially
   b) Non-contiguously
   c) Dynamically
   d) None of the above

---

2. File Handling

6. Which function is used to open a file in C?
   a) fopen()
   b) open()
   c) read()
   d) write()
7. What does the mode `"w+"` mean in the `fopen()` function?
   a) Read-only
   b) Write and overwrite
   c) Read and write
   d) Append
8. Which function is used to write a single character to a file?
   a) putchar()
   b) fwrite()

c) fputc()
d) fprintf()

9. What does `fseek()` function do?
   a) Reads from a file
   b) Moves the file pointer
   c) Writes to a file
   d) Closes a file
10. Which function is used to get the current position of the file pointer?
    a) fpos()
    b) ftell()
    c) fgetpos()
    d) None of the above

---

3. Image Processing

11. What is the typical format for storing raw image data in C?
    a) BMP
    b) ASCII
    c) RGB arrays
    d) None of the above
12. Which library is commonly used for image manipulation in C?
    a) SDL
    b) OpenCV
    c) libjpeg
    d) All of the above
13. How can you read pixel data from an image in C?
    a) fread()
    b) fgetc()
    c) fscanf()
    d) fwrite()
14. What does a grayscale image typically store?
    a) RGB values
    b) Intensity values
    c) Binary values
    d) None of the above
15. What is the purpose of histogram equalization in image processing?
    a) Resizing an image
    b) Improving contrast
    c) Compressing an image
    d) Removing noise

---

4. Advanced Basics

What is the output of the following code?

```
printf("%d", sizeof('A'));
```

16. a) 1
    b) 2
    c) 4
    d) Compiler error
17. Which of the following operators has the highest precedence in C?
    a) *
    b) +
    c) ()
    d) &&
18. What does the `volatile` keyword indicate in C?
    a) A variable can change unexpectedly
    b) A variable is static
    c) A variable is constant
    d) A variable is global
19. What will happen if you declare `extern int x;` without defining `x`?
    a) Compilation error
    b) No error
    c) Runtime error
    d) None of the above
20. Which data type is best suited for storing a pointer in C?
    a) void *
    b) int *
    c) char *
    d) Any of the above

---

5. Functions

21. What does the `return` statement do in a C function?
    a) Terminates a function
    b) Returns a value
    c) Both a and b
    d) None of the above
22. How can you pass an array to a function?
    a) By value
    b) By reference
    c) By both
    d) None of the above

23. What is the purpose of a function prototype in C?
    a) To define a function
    b) To declare a function
    c) To call a function
    d) None of the above
24. Can a function in C return multiple values?
    a) Yes, using an array or structure
    b) Yes, using pointers
    c) No
    d) Both a and b

What is the output of the following code?

```
void test() {
       printf("Hello");
   }
   int main() {
       printf("%d", sizeof(test));
   }
```

25. a) 1
    b) 2
    c) 4
    d) Compiler error

---

6. Arrays, Pointers, and DMA

What is the output of the following code?

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

26. a) 1
    b) 2
    c) 3
    d) Compiler error
27. Which of the following is true for pointer arithmetic?
    a) Only addition is allowed
    b) Both addition and subtraction are allowed
    c) Multiplication is allowed
    d) None of the above

28. What does `malloc()` return when memory allocation fails?
    a) -1
    b) NULL
    c) Garbage value
    d) None of the above
29. Which function is used to deallocate memory allocated by `malloc()`?
    a) free()
    b) delete()
    c) clear()
    d) None of the above
30. What is the advantage of using `calloc()` over `malloc()`?
    a) It initializes memory to zero
    b) It is faster
    c) It requires less memory
    d) None of the above