Midterm

1. What is the purpose of stdin and stdout?

A) File handling
B) Input and output streams
C) Memory allocation
D) Pointer initialization

_____

2. Which of the following correctly declares a pointer to an integer?

A) int p;
B) int *p;
C) int &p;
D) pointer int p;

_____

3. Which bitwise operator in C is used to toggle (flip) bits?

A) &
B) |
C) ^
D) ~

_____

4. What happens if you free() memory twice?

A) It safely clears memory
B) Causes a segmentation fault (undefined behaviour)
C) Reallocates memory
D) Has no effect

_____

5. Which of the following statements about arrays in C is false?

A) Array indices start at 0
B) Arrays can be dynamically resized
C) Array name represents the address of the first element
D) Arrays can store only same data types

_____

6. The statement int x; x = 5; defines x as:

A) A constant variable
B) A global variable
C) A local integer variable
D) A static variable

_____

7. What is the purpose of break in loops?

A) Skip an iteration
B) Exit the loop immediately
C) End loop after running the code once
D) End the program

_____

8. Which of the following statements is false about switch?

A) It can test equality of integers and characters
B) It supports ranges directly
C) Each case must end with break to avoid fall-through
D) It's faster than a long series of if-else

_____

9. Arrays in C are:

A) Dynamic data structures
B) Always passed by value
C) Passed as pointers to functions
D) None of the above

_____

10. The safest way to find an array's length in bytes is:

A) sizeof(arr)

B) sizeof(arr) / sizeof(arr[0])
C) sizeof(arr[0])
D) length(arr)

_____

11. Which function reads a line including whitespace in C?

A) scanf("%s", str);
B) gets(str);
C) fgets(str, size, stdin);
D) getchar();

_____

12. In scanf("%d", &x);, why is & used?

A) To dereference the value
B) To access variable's address
C) To perform AND operation
D) To multiply input by 2

_____

13. What will printf("%d\n", 'A'); output?

A) A
B) ASCII value of A (65)
C) Error
D) 0

_____

14. What is true about arrays and pointers?

A) Array names are pointers to their first element
B) They can be assigned directly (arr1 = arr2;)
C) Both are identical in all cases
D) Pointers cannot be used for arrays

_____

15. Recursion mainly uses which data structure internally?

A) Queue
B) Stack
C) Heap
D) Tree

_____

16. Which of the following is true about the main() function in C?

A) It must always return 0
B) It can return any integer to indicate exit status
C) It must take no arguments
D) It cannot contain local variables

_____

17. In C, what does the array name arr represent?

A) The address of the array
B) The value of the first element
C) A constant pointer to the first element
D) A modifiable pointer to the array

_____

18. A string in C ends when:

A) Compiler detects whitespace
B) Memory page ends
C) Null character '\0' is encountered
D) The length reaches 255

_____

19. Each recursive call creates:

A) A copy of the entire program
B) A new frame in the call stack
C) A new pointer to the same memory
D) A loop with fewer iterations

———

20. According to DeMorgan's Law: ~(A | B) is equivalent to:

A) ~A | ~B
B) ~A & ~B
C) A & B
D) A ^ B

———

21. When you pass an array to a function:

A) The entire array is copied
B) Only a pointer to its first element is passed
C) Both array and pointer are passed
D) Compiler decides based on optimization

———

22. What does typedef struct { int x, y; } point_t; do?

A) Declares one variable point_t
B) Creates a new type alias point_t
C) Defines a global function
D) Allocates memory for points

———

23. In char *p = "Hi";, what does p store?

A) The first character only
B) The address of the first character
C) The length of the string
D) The entire string directly

———

24. Which of the following correctly defines a string?

A) char s = "Hello";
B) char s[] = "Hello";

C) string s = "Hello";
D) char *s = ‘Hello';

25. What is the output of:

printf("%d", printf("%d", printf("%d", 0)));

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

Part 2: Code Questions (25 marks)

Q1 (5 points)

On a 64-bit system where sizeof(int) = 4 and pointers are 8 bytes, what prints?

```
#include <stdio.h>

void g(int a[]) {   // actually int *a
    printf("%zu ", sizeof(a));
}

int main() {
    int a[10] = {0};
    printf("%zu ", sizeof(a));
    g(a);
    printf("%zu\n", sizeof(&a[0]));
}
```

Q2 (5 points)

What would the following code output?

```
#include <stdio.h>

void test(int *p) {
    *p = *p + 5;
    printf("%d ", *p);
}
int main() {
```

```
    int x = 10;
    test(&x);
    printf("%d", x);
    return 0;
}
```

_____

Q3 (5 points)

Complete the code to get the expected output:

```
#include <stdio.h>

int main() {
    int arr[] = {3, 6, 8, 11, 14, 17};
    int n = sizeof(arr) / sizeof(arr[0]);
    int sum = 0;
    int *p;

    for (p = arr; p < arr + n; p++) {
        if (____________) {
            sum += _______;
        }
    }

    printf("Sum = %d\n", sum);
    return 0;
}
```

Hint: Sum of even numbers (6 + 8 + 14)
Expected Output: Sum = 28

Q4 (10 points)
Tower of Hanoi is a mathematical puzzle where we have three rods (A, B, and C) and N disks.

- Only one disk can be moved at a time
- A disk can only be moved if it is the uppermost disk
- No disk may be placed on top of a smaller disk

Write a Tower of Hanoi program that takes input (number of disks) and prints the solution using recursion.