Top 10 c interview questions

1️⃣Difference between malloc() and calloc()

Answer:

malloc() → allocates memory, does not initialize

calloc() → allocates memory, initializes with zero

2️⃣What is a pointer? Why do we use pointers?

Answer: Pointers store memory addresses and are used for:

Dynamic memory allocation

Passing arguments efficiently

Working with arrays and strings

3️⃣Difference between ++i and i++

Answer:

++i → increment first, then use

i++ → use first, then increment

4️⃣What is static keyword in C?

Answer:

Keeps value between function calls

Limits variable scope to the same file

5️⃣What is the difference between struct and union?

Answer:

struct → each member has separate memory

union → all members share the same memory

6️⃣What is segmentation fault?

Answer: A segmentation fault occurs when a program tries to access invalid memory, like using an uninitialized pointer.

7️⃣Difference between array and pointer

Answer:

Array → fixed size, memory allocated at compile time

Pointer → can point to different memory locations

8️⃣What is const keyword?

Answer: const makes a variable read-only (value cannot be changed).

9️⃣What is recursion? Give an example

Answer: Recursion is when a function calls itself to solve a problem.

🔟 Difference between break and exit()

Answer:

break → exits loop only

exit() → terminates the entire program

Storage Classes in C

Introduction

In C programming, storage classes define the scope, lifetime, visibility, and memory location of variables and functions.
They control where a variable is stored, how long it exists, and which parts of the program can access it.

C provides four storage classes:

1. auto

2. register

3. static

4. extern

1. auto Storage Class

The auto storage class is the default storage class for all local variables.

Features:

Scope: Inside the block where it is declared

Lifetime: Till the execution of the block

Memory location: Stack

Initial value: Garbage value

Example:

```
void main() {
    auto int x = 10;
}
```

👉 Note: Writing auto is optional because local variables are auto by default.

2. register Storage Class

The register storage class is used to store variables in CPU registers instead of RAM for faster access.

Features:

Scope: Local to the block

Lifetime: Till block execution

Memory location: CPU register (if available)

Cannot access address using &

Example:

```
void main() {
    register int count;
}
```

👉 Note: It is only a request to the compiler. The compiler may ignore it.

3. static Storage Class

The static storage class preserves the value of a variable between function calls.

Features:

Scope: Local to function or file

Lifetime: Entire program execution

Memory location: Data segment

Initial value: Zero (by default)

Example:

```
void fun() {
    static int count = 0;
    count++;
    printf("%d ", count);
}
```

👉 Output for multiple calls: 1 2 3

4. extern Storage Class

The extern storage class is used to access global variables defined in another file.

Features:

Scope: Global

Lifetime: Entire program execution

Memory location: Data segment

Does not allocate memory (only references it)

Example:

```
extern int total;
```

👉 Used mainly in multi-file programs.