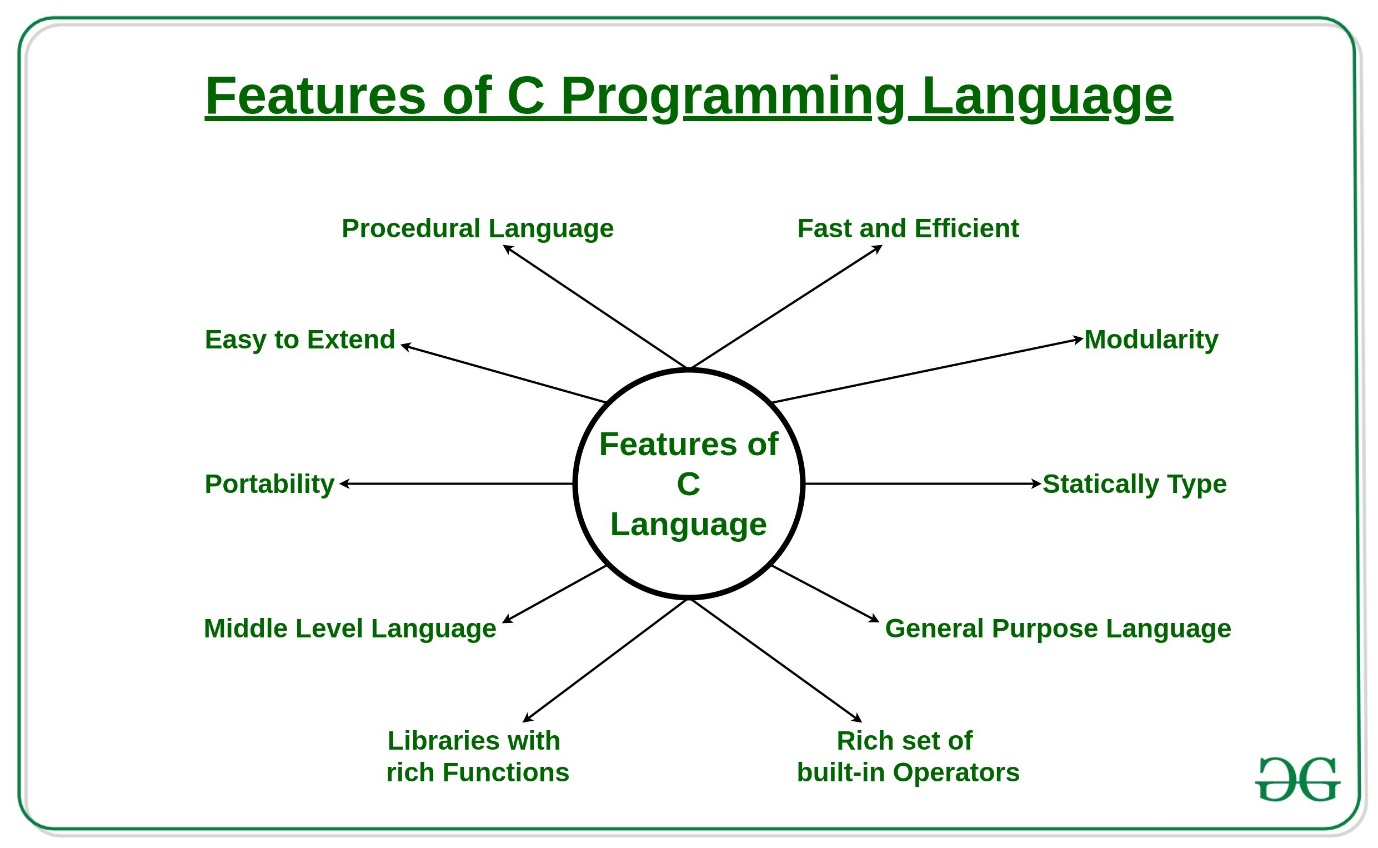
1. **What are the features of the C programming language?**

****

### What are basic data types supported in the C Programming Language?

### A diagram of data types Description automatically generated

### 3. What is the use of static variables in C?

Static variables in the C programming language are used to preserve the data values between function calls even after they are out of their scope. Static variables preserve their values in their scope and they can be used again in the program without initializing again. Static variables have an initial value assigned to 0 without initialization.

A screenshot of a computer program

Description automatically generated

A close-up of a white box

Description automatically generated

### 4. Write a program to convert a number to a string with the help of sprintf() function in the C library.

### 

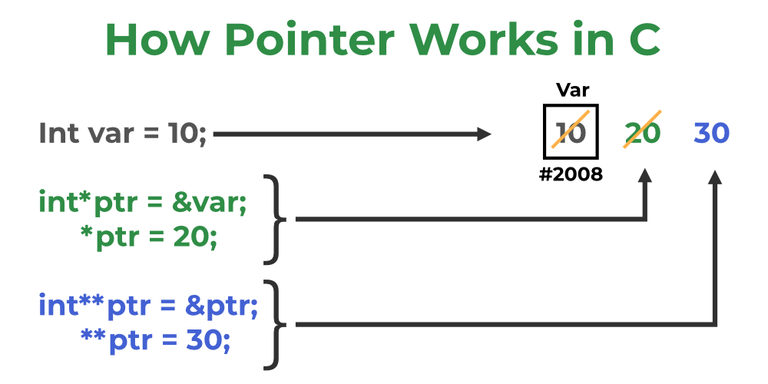
### 5. What is recursion in C?

Recursion is the process of making the function call itself directly or indirectly. A recursive function solves a particular problem by calling a copy of itself and solving smaller subproblems that sum up the original problems. Recursion helps to reduce the length of code and make it more understandable. The recursive function uses a LIFO ( Last In First Out ) structure like a [stack](https://www.geeksforgeeks.org/stack-data-structure/). Every recursive call in the program requires extra space in the stack memory.

**6. What are pointers and their uses?**

Pointers are used to store the address of the variable or a memory location. Pointer can also be used to refer to another pointer function. The main purpose of the pointer is to save memory space and increase execution time. Uses of pointers are:

* To pass arguments by reference
* For accessing array elements
* To return multiple values
* Dynamic memory allocation
* To implement data structures
* To do system-level programming where memory addresses are useful



### 7. What is typedef in C?

In C programming, typedef is a keyword that defines an alias for an existing type. Whether it is an integer variable, function parameter, or structure declaration, typedef will shorten the name.

**Syntax:**

*typedef <existing-type> <alias-name>*

**Example:**

typedef long long ll

**8. What is the difference between type casting and type conversion?**

| **Type Casting** | **Type Conversion** |
| --- | --- |
| The data type is converted to another data type by a programmer with the help of a casting operator. | The data type is converted to another data by a compiler. |

**9. What are the functions and their types?**

The function is a block of code that is used to perform a task multiple times rather than writing it out multiple times in our program. Functions avoid repetition of code and increase the readability of the program. Modifying a program becomes easier with the help of function and hence reduces the chances of error. *There are two types of functions:*

* **User-defined Functions:** Functions that are defined by the user to reduce the complexity of big programs. They are built only to satisfy the condition in which the user is facing issues and are commonly known as “tailor-made functions”.
* **Built-in Functions:** Library functions are provided by the compiler package and consist of special functions with special and different meanings. These functions give programmers an edge as we can directly use them without defining them.

**10. What is the difference between macro and functions?**

* A macro is a name that is given to a block of C statements as a pre-processor directive. Macro is defined with the pre-processor directive. Macros are pre-processed which means that all the macros would be preprocessed before the compilation of our program. However, functions are not preprocessed but compiled.

| **Macro** | **Function** |
| --- | --- |
| Macros are preprocessed. | Functions are compiled. |
| Code length is increased using macro. | Code length remains unaffected using function. |
| Execution speed using a macro is faster. | Execution speed using function is slower. |

**11. How to convert a string to numbers in C?**

In C we have 2 main methods to convert strings to numbers i.e, Using string stream, Using stoi() library Function, and using atoi() library function.

* **sscanf():** It reads input from a string rather than standard input.
* **atoi() or stoi():** These functions takes a string literal or a character array as an argument and an integer value is returned.

**12. What is the difference between call by value and call by reference?**

| **Call by value** | **Call by Reference** |
| --- | --- |
| Values of the variable are passed while function calls. | The address of a variable(location of variable) is passed while the function call. |
| Dummy variables copy the value of each variable in the function call. | Dummy variables copy the address of actual variables. |
| Changes made to dummy variables in the called function have no effect on actual variables in the calling function. | We can manipulate the actual variables using addresses. |

### 13. What is the sleep() function?

sleep() function in C allows the users to wait for a current thread for a given amount of time. sleep() function will sleep the present executable for the given amount of time by the thread but other operations of the CPU will function properly. sleep() function returns 0 if the requested time has elapsed.

### 14. Write a C program to print the Fibonacci series using recursion and without using recursion.

### 

A screenshot of a computer code

Description automatically generated

### A computer screen shot of a code Description automatically generated

**15. What is static memory allocation and dynamic memory allocation?**

* **Static memory allocation:**Memory allocation which is done at compile time is known as static memory allocation. Static memory allocation saves running time. It is faster than dynamic memory allocation as memory allocation is done from the stack. This memory allocation method is less efficient as compared to dynamic memory allocation. It is mostly preferred in the array.
* **Dynamic memory allocation:**Memory allocation done at execution or run time is known as dynamic memory allocation. Dynamic memory allocation is slower than static memory allocation as memory allocation is done from the heap. This memory allocation method is more efficient as compared to static memory allocation. It is mostly preferred in the linked list.

DS-----------------------------------------------------------------------------

### ****16: What is the time complexity for accessing an element in an array?****

**Answer:**The[time complexity](https://www.geeksforgeeks.org/understanding-time-complexity-simple-examples/) for accessing an element in an array is**O(1),** as it can be accessed directly using its index.

### ****17: How can you find duplicate elements in an array?****

**Answer:**One way to find duplicate elements in an array is to use a [hash set](https://www.geeksforgeeks.org/hashset-in-java/) or to sort the array and then iterate through it to find consecutive duplicates.

**18: What is time complexity of Linked List operations?**

**Answer:**The time complexity of common operations on a singly-linked list are as follows:

**Insertion:**

* At the beginning: O(1)
* At the end: O(n)
* At a specific position: O(n)

**Deletion:**

* At the beginning: O(1)
* At the end: O(n)
* At a specific position: O(n)

**Search:**O(n)  
**Traversal:** O(n)

### ****19:**** What is a stack underflow?

**Answer:**A stack underflow occurs when the stack is empty and an attempt is made to pop an element.

**20: What are the different types of Queues?**

**Answer:**

* [Simple Queue](https://www.geeksforgeeks.org/introduction-to-queue-data-structure-and-algorithm-tutorials/)
* [Circular Queue](https://www.geeksforgeeks.org/introduction-to-circular-queue/)
* [Priority Queue](https://www.geeksforgeeks.org/priority-queue-set-1-introduction/)
* [Double-Ended Queue (Deque)](https://www.geeksforgeeks.org/deque-set-1-introduction-applications/)

### ****21:**** What is a heap data structure?

**Answer:**A[heap](https://www.geeksforgeeks.org/heap-data-structure/) is a complete[binary tree](https://www.geeksforgeeks.org/binary-tree-data-structure/) that satisfies the heap property: each node’s value is greater than or equal to its children’s values.

### ****22:**** What are the two types of heaps?

**Answer:** [Max-heap](https://www.geeksforgeeks.org/introduction-to-max-heap-data-structure/) and[min-heap](https://www.geeksforgeeks.org/introduction-to-min-heap-data-structure/). In a max-heap, the root node has the maximum value, while in a min-heap, the root node has the minimum value.

**23: What are the applications of heaps?**

**Answer: Heap applications:**

* [Priority queues](https://www.geeksforgeeks.org/priority-queue-set-1-introduction/)
* [Sorting](https://www.geeksforgeeks.org/sorting-algorithms/)
* Finding the median
* [Implementing Dijkstra’s algorithm](https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/)
* Network routing
* [Huffman coding](https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/)

**24: What is the difference between a heap and a binary search tree (BST)?**

**Answer:** A heap is a complete binary tree that satisfies the heap property, while a BST is a partially ordered binary tree that satisfies the BST property.

### ****25****: Explain how a hash function works.

**Answer:**A hash function takes an input key and maps it to a fixed-size index (hash value) within the hash table’s array. Ideally, the function distributes keys evenly across the array to minimize collisions. Common hash functions include **modulo division, bitwise operations,**and**polynomial hashing**.

**26: Explain different types of trees.**

**Answer:**

* [Binary Tree:](https://www.geeksforgeeks.org/binary-tree-data-structure/) Each node has at most two children (left and right).
* [Full Binary Tree:](https://www.geeksforgeeks.org/full-binary-tree/)Every node except leaves has two children.
* [Complete Binary Tree:](https://www.geeksforgeeks.org/complete-binary-tree/)All levels are filled except possibly the last, and nodes are filled left to right.
* [Perfect Binary Tree:](https://www.geeksforgeeks.org/perfect-binary-tree/)Every node has two children, and all leaves are at the same level.
* [AVL Tree:](https://www.geeksforgeeks.org/introduction-to-avl-tree/)Self-balancing binary search tree with a height difference of at most 1 between subtrees.
* [Red-Black Tree:](https://www.geeksforgeeks.org/introduction-to-red-black-tree/)Self-balancing binary search tree with specific coloring rules to maintain balance.
* [B-Tree:](https://www.geeksforgeeks.org/introduction-of-b-tree-2/)Generalization of a binary search tree with more than two children per node.

**27: What are the basic operations performed on a tree?**

**Answer:**

* **Insertion:**Add a new node to the tree while maintaining its properties (e.g., ordering in search trees).
* **Deletion:** Remove a node from the tree while preserving its structure.
* **Traversal:** Visit each node in the tree exactly once in a specific order (preorder, inorder, postorder).
* **Searching:** Find a specific node with a given value based on search criteria.

### ****28:**** When would you choose a tree over other data structures like arrays or linked lists?

**Answer:**Trees are ideal for hierarchical data, maintaining relationships between elements, and efficient searching based on order. Arrays or linked lists are better for simple linear data or frequent insertions/deletions at specific positions.