

1. Define Array and Its Types

An array is a data structure used to store a collection of elements of the same data type in a continuous memory location. Each element in an array is accessed using an index value. Arrays allow easy storage and retrieval of data using a single variable name. Arrays are mainly of two types. One-dimensional arrays store data in a linear form. Multidimensional arrays store data in more than one dimension, such as rows and columns. Arrays are widely used due to their simplicity and fast access time.

2. Multidimensional Array

A multidimensional array is an array that has more than one dimension. It is mainly used to store data in a structured form such as tables or matrices. It helps in organizing large amounts of related data in rows and columns, making data processing easier and more systematic.

3. Array Initialization Methods

Array initialization is the process of assigning values to an array. One method is static initialization, where values are assigned at the time of declaration. Another method is dynamic initialization, where values are assigned during program execution. Both methods help in storing data efficiently depending on program requirements.

4. Sorting Methods

Sorting is the process of arranging data in a specific order such as ascending or descending. There are many sorting methods used in data structures. Simple sorting methods compare and arrange elements step by step. Advanced sorting methods divide data into smaller parts and then combine them in sorted order. Sorting improves data searching and overall program efficiency.

5. Program to Sort Elements of a Linked List

The following program sorts elements of a linked list using a simple comparison-based approach.

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node* next;
};

void sortList(struct Node* head) {
    struct Node *i, *j;
    int temp;
    for(i = head; i != NULL; i = i->next) {
        for(j = i->next; j != NULL; j = j->next) {
            if(i->data > j->data) {
                temp = i->data;
                i->data = j->data;
                j->data = temp;
            }
        }
    }
}
```

6. Difference Between Array and Linked List

An array stores elements in continuous memory locations, while a linked list stores elements in separate memory locations connected using links. Arrays have fixed size, whereas linked lists can grow or shrink dynamically. Accessing elements in arrays is faster, while insertion and deletion operations are easier in linked lists.

7. Operations on Linked List

Linked lists support various operations. These include insertion of elements, deletion of elements, traversal to display data, searching for an element, and updating values. These operations help in managing data efficiently and flexibly.

8. Advantages of Linked List Over Array

Linked lists have several advantages compared to arrays. They allow dynamic memory allocation. Insertion and deletion operations are easier and faster. Memory is utilized efficiently as it is allocated only when required. Linked lists are suitable for applications where data size changes frequently.