Array Insertion

🧠 INSERTION IN ARRAY (C & C++)

🔷 What is Insertion?

Insertion means adding a new element into an array —

either at the **beginning**, **end**, or at a **specific position**.

Since arrays have **fixed size**, we must **shift elements right** to make space for the new one.

🔷 Types of Insertion

1. **Insertion at Beginning**
2. **Insertion at End**
3. **Insertion at a Specific Position (Index)**
4. **Insertion After a Given Value**

🧠 1️⃣ Insertion at the Beginning

🔶 Algorithm

1. Input the array and its size `n`.
2. Input the new element `x` to insert.
3. Shift all elements one step to the right (from last to first).
4. Place `x` at index `0`.
5. Increase `n` by 1.

🔶 C Code

```c
#include <stdio.h>

int main() {
    int arr[100], n, x, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter elements: ");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Enter element to insert at beginning: ");
    scanf("%d", &x);

    // Shift elements to right
    for (i = n; i > 0; i--) {
        arr[i] = arr[i - 1];
    }

    arr[0] = x;
    n++;

    printf("Array after insertion: ");
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

🔶 C++ Code

```
#include <iostream>
using namespace std;

int main() {
    int arr[100], n, x;

    cout << "Enter number of elements: ";
    cin >> n;

    cout << "Enter elements: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    cout << "Enter element to insert at beginning: ";
    cin >> x;

    // Shift elements to right
    for (int i = n; i > 0; i--) {
        arr[i] = arr[i - 1];
    }

    arr[0] = x;
    n++;

    cout << "Array after insertion: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    return 0;
}
```

🧠 2️⃣ Insertion at the End

🔶 Algorithm

1. Input the array and its size `n`.
2. Input the new element `x`.
3. Place the element at index `n`.
4. Increase `n` by 1.

🔶 C Code

```
#include <stdio.h>

int main() {
    int arr[100], n, x, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter elements: ");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Enter element to insert at end: ");
    scanf("%d", &x);

    arr[n] = x;
    n++;

    printf("Array after insertion: ");
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

🔶 C++ Code

```
#include <iostream>
using namespace std;

int main() {
    int arr[100], n, x;

    cout << "Enter number of elements: ";
    cin >> n;

    cout << "Enter elements: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    cout << "Enter element to insert at end: ";
    cin >> x;

    arr[n] = x;
    n++;

    cout << "Array after insertion: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    return 0;
}
```

🧠 3️⃣ Insertion at a Specific Position (Index-Based)

🔶 Algorithm

1. Input array and its size `n`.

2. Input the new element `x` and the position `pos` (0-based index).
3. Check if `pos` is valid (`0 <= pos <= n`).
4. Shift all elements right from index `n-1` to `pos`.
5. Place the new element at index `pos`.
6. Increase `n` by 1.

🔶 C Code

```
#include <stdio.h>

int main() {
    int arr[100], n, x, pos, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter elements: ");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Enter position to insert (0-based index): ");
    scanf("%d", &pos);

    printf("Enter element to insert: ");
    scanf("%d", &x);

    if (pos < 0 || pos > n) {
        printf("Invalid position!\n");
        return 0;
    }

    for (i = n; i > pos; i--) {
        arr[i] = arr[i - 1];
    }
```

```
    arr[pos] = x;
    n++;

    printf("Array after insertion: ");
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

🔶 C++ Code

```
#include <iostream>
using namespace std;

int main() {
    int arr[100], n, x, pos;

    cout << "Enter number of elements: ";
    cin >> n;

    cout << "Enter elements: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    cout << "Enter position to insert (0-based index): ";
    cin >> pos;

    cout << "Enter element to insert: ";
    cin >> x;

    if (pos < 0 || pos > n) {
        cout << "Invalid position!" << endl;
        return 0;
```

```
    }

    for (int i = n; i > pos; i--) {
        arr[i] = arr[i - 1];
    }

    arr[pos] = x;
    n++;

    cout << "Array after insertion: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    return 0;
}
```

🧠 4️⃣ Insertion After a Given Value

🔶 Algorithm

1. Input array and size `n`.
2. Input the target value `val` and new element `x`.
3. Find index of `val` in array.
4. Shift elements right from `pos + 1` onwards.
5. Insert new element `x` at `pos + 1`.
6. Increase `n` by 1.
7. If `val` not found, print “Value not found”.

🔶 C Code

```c
#include <stdio.h>

int main() {
    int arr[100], n, x, val, pos = -1, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter elements: ");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Enter value after which to insert: ");
    scanf("%d", &val);

    printf("Enter element to insert: ");
    scanf("%d", &x);

    for (i = 0; i < n; i++) {
        if (arr[i] == val) {
            pos = i;
            break;
        }
    }

    if (pos == -1) {
        printf("Value not found!\n");
        return 0;
    }

    for (i = n; i > pos + 1; i--) {
        arr[i] = arr[i - 1];
    }

    arr[pos + 1] = x;
```

```
    n++;

    printf("Array after insertion: ");
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

🔶 C++ Code

```
#include <iostream>
using namespace std;

int main() {
    int arr[100], n, x, val, pos = -1;

    cout << "Enter number of elements: ";
    cin >> n;

    cout << "Enter elements: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    cout << "Enter value after which to insert: ";
    cin >> val;

    cout << "Enter element to insert: ";
    cin >> x;

    for (int i = 0; i < n; i++) {
        if (arr[i] == val) {
            pos = i;
            break;
        }
```

```
    }

    if (pos == -1) {
        cout << "Value not found!" << endl;
        return 0;
    }

    for (int i = n; i > pos + 1; i--) {
        arr[i] = arr[i - 1];
    }

    arr[pos + 1] = x;
    n++;

    cout << "Array after insertion: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    return 0;
}
```

🧩 Example Execution

Input:

```
5
10 20 30 40 50
Insert 25 at position 2
```

Output:

```
10 20 25 30 40 50
```

🧮 Time Complexity Summary

Type of Insertion	Description	Time Complexity
Beginning	Shift all elements right	**O(n)**
End	Direct insertion	**O(1)**
Specific Position	Shift elements right from index	**O(n)**
After Given Value	Search + Shift	**O(n)**