- **Binary Search**

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

int binarySearch(int arr[], int size, int target) {
    int left = 0;
    int right = size - 1;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        // Check if the target is present at the middle
        if (arr[mid] == target) {
            return mid;
        }

        // If the target is greater, ignore the left half
        if (arr[mid] < target) {
            left = mid + 1;
        }

        // If the target is smaller, ignore the right half
        else {
            right = mid - 1;
        }
    }

    // If we reach here, then the element was not present
    return -1;
}

int main() {
    // Example usage
    int sortedArray[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    int size = 10;  // Explicitly specify the size of the array
    int target;
    cout << " Enter num of search in array ";
    cin >> target;

    int result = binarySearch(sortedArray, size, target);

    if (result != -1) {
        std::cout << "Element found at index " << result << std::endl;
    }
    else {
        std::cout << "Element not found in the array" << std::endl;
    }

    return 0;
}
```

- **Selection SORT**

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

void swap(int& a, int& b) {
    int temp = a;
    a = b;
    b = temp;
}

void selectionSort(int arr[], int size) {
    for (int i = 0; i < size - 1; ++i) {
        int minIndex = i;

        // Find the index of the minimum element in the unsorted part
        for (int j = i + 1; j < size; ++j) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j;
            }
        }

        // Swap the found minimum element with the element at the current index
        if (minIndex != i) {
            swap(arr[i], arr[minIndex]);
        }
    }
}

int main() {
    int arr[5] = { 64, 25, 12, 22, 11 };
    int size = 5;  // Specify the size directly

    cout << "Original array: ";
    for (int i = 0; i < size; ++i) {
        cout << arr[i] << " ";
    }

    selectionSort(arr, size);

    cout << "\nSorted array: ";
    for (int i = 0; i < size; ++i) {
        cout << arr[i] << " ";
    }

    return 0;
}
```