# Selection Sort

```
#include <bits/stdc++.h>
using namespace std;

// Function for Selection sort
void selectionSort(int arr[], int n)
{
	int i, j, min_idx;

	// One by one move boundary of
	// unsorted subarray
	for (i = 0; i < n - 1; i++) {

		// Find the minimum element in
		// unsorted array
		min_idx = i;
		for (j = i + 1; j < n; j++) {
			if (arr[j] < arr[min_idx])
				min_idx = j;
		}

		// Swap the found minimum element
		// with the first element
		if (min_idx != i)
			swap(arr[min_idx], arr[i]);
	}
}

// Function to print an array
void printArray(int arr[], int size)
{
	int i;
	for (i = 0; i < size; i++) {
		cout << arr[i] << " ";
		cout << endl;
	}
}

int main()
{
	int arr[] = { 64, 25, 12, 22, 11 };
	int n = sizeof(arr) / sizeof(arr[0]);

	// Function Call
	selectionSort(arr, n);
	cout << "Sorted array: \n";
printArray(arr, n);
	return 0;
}
```

# Bubble Sort

```
#include <bits/stdc++.h>
using namespace std;

// An optimized version of Bubble Sort
void bubbleSort(int arr[], int n)
{
	int i, j;
	bool swapped;
	for (i = 0; i < n - 1; i++) {
		swapped = false;
		for (j = 0; j < n - i - 1; j++) {
			if (arr[j] > arr[j + 1]) {
				swap(arr[j], arr[j + 1]);
				swapped = true;
			}
		}

		// If no two elements were swapped
		// by inner loop, then break
		if (swapped == false)
			break;
	}
}

// Function to print an array
void printArray(int arr[], int size)
{
	int i;
	for (i = 0; i < size; i++)
		cout << " " << arr[i];
}

// Driver program to test above functions
int main()
{
	int arr[] = { 64, 34, 25, 12, 22, 11, 90 };
	int N = sizeof(arr) / sizeof(arr[0]);
	bubbleSort(arr, N);
	cout << "Sorted array: \n";
	printArray(arr, N);
	return 0;
}
```