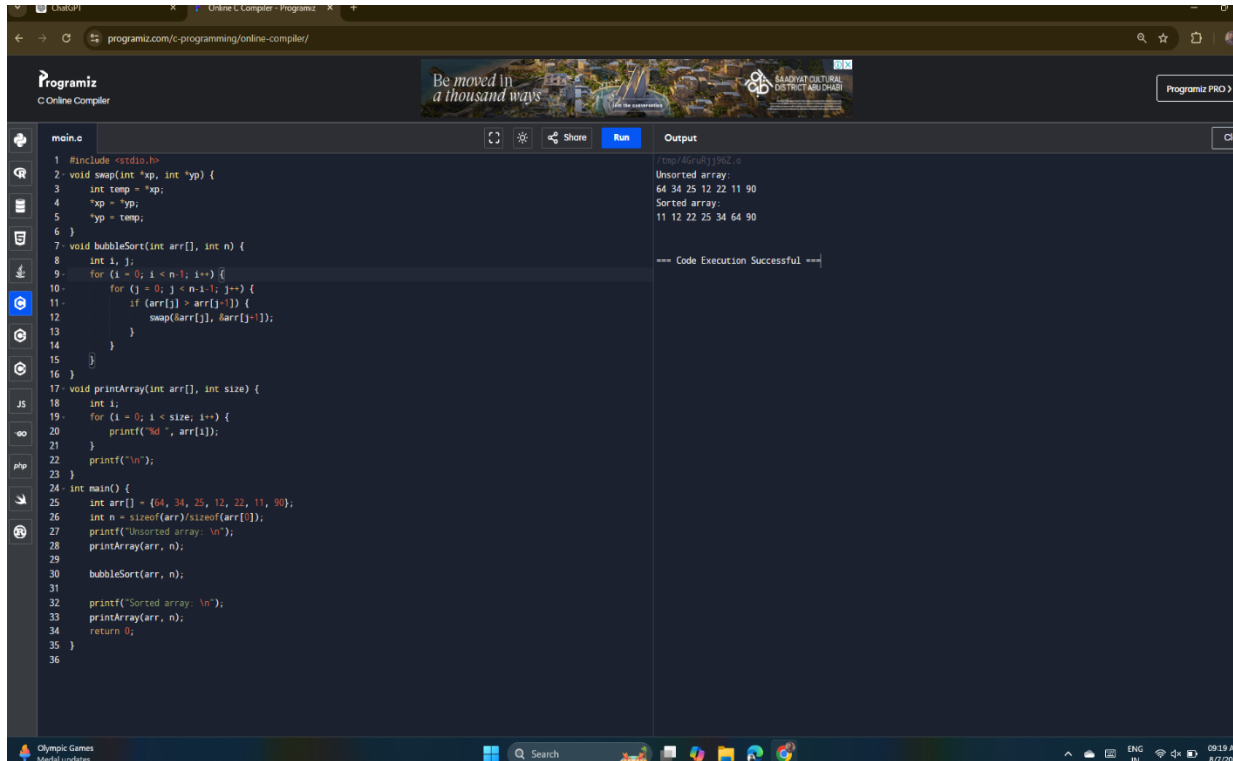


## 1. Bubble sorting code implementation in c language



The screenshot shows the Programiz C Online Compiler interface. The code editor on the left contains the following C code for Bubble Sort:

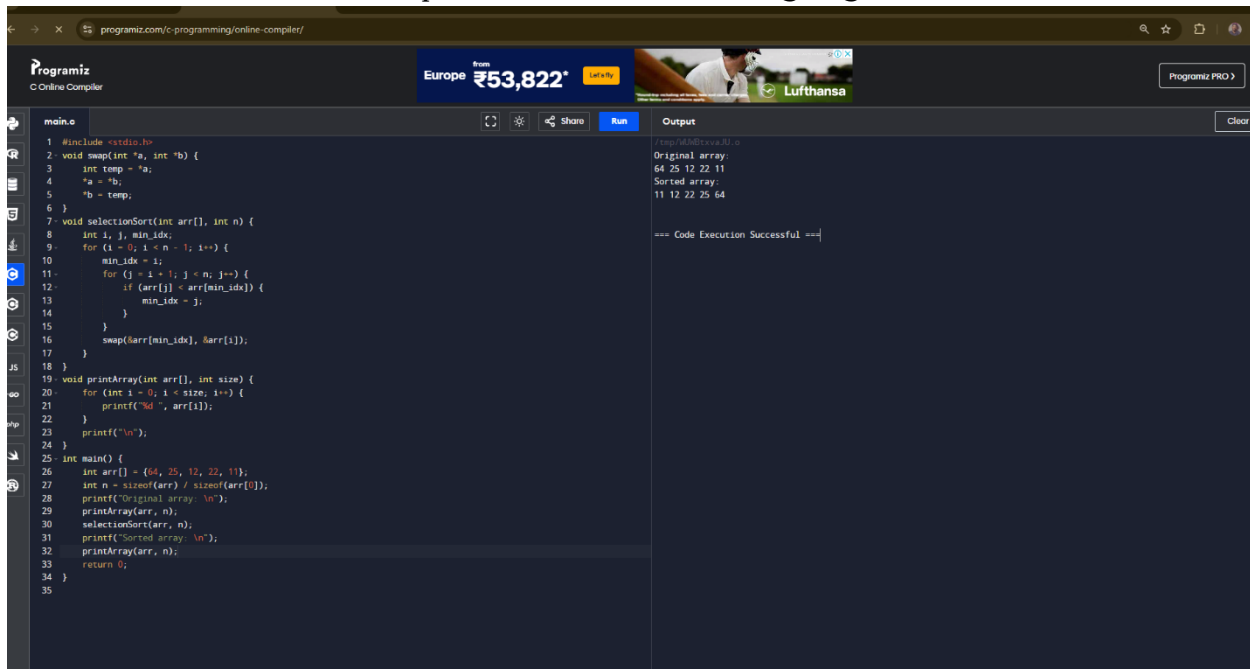
```
1 #include <stdio.h>
2 void swap(int *xp, int *yp) {
3     int temp = *xp;
4     *xp = *yp;
5     *yp = temp;
6 }
7 void bubbleSort(int arr[], int n) {
8     int i, j;
9     for (i = 0; i < n-1; i++) {
10        for (j = 0; j < n-1-i; j++) {
11            if (arr[j] > arr[j+1]) {
12                swap(&arr[j], &arr[j+1]);
13            }
14        }
15    }
16 }
17 void printArray(int arr[], int size) {
18     int i;
19     for (i = 0; i < size; i++) {
20         printf("%d ", arr[i]);
21     }
22     printf("\n");
23 }
24 int main() {
25     int arr[] = {64, 34, 25, 12, 22, 11, 90};
26     int n = sizeof(arr)/sizeof(arr[0]);
27     printf("Unsorted array: \n");
28     printArray(arr, n);
29     bubbleSort(arr, n);
30     printf("Sorted array: \n");
31     printArray(arr, n);
32     return 0;
33 }
```

The output window on the right shows the execution results:

```
Unsorted array:
64 34 25 12 22 11 90
Sorted array:
11 12 22 25 34 64 90

=== Code Execution Successful ===
```

## 2. Selection Sort code implementation in c language



The screenshot shows the Programiz C Online Compiler interface. The code editor on the left contains the following C code for Selection Sort:

```
1 #include <stdio.h>
2 void swap(int *a, int *b) {
3     int temp = *a;
4     *a = *b;
5     *b = temp;
6 }
7 void selectionSort(int arr[], int n) {
8     int i, j, min_idx;
9     for (i = 0; i < n-1; i++) {
10        min_idx = i;
11        for (j = i+1; j < n; j++) {
12            if (arr[j] < arr[min_idx]) {
13                min_idx = j;
14            }
15        }
16        swap(&arr[min_idx], &arr[i]);
17    }
18 }
19 void printArray(int arr[], int size) {
20     for (int i = 0; i < size; i++) {
21         printf("%d ", arr[i]);
22     }
23     printf("\n");
24 }
25 int main() {
26     int arr[] = {64, 25, 12, 22, 11};
27     int n = sizeof(arr) / sizeof(arr[0]);
28     printf("Original array: \n");
29     printArray(arr, n);
30     selectionSort(arr, n);
31     printf("Sorted array: \n");
32     printArray(arr, n);
33     return 0;
34 }
```

The output window on the right shows the execution results:

```
Original array:
64 25 12 22 11
Sorted array:
11 12 22 25 64

=== Code Execution Successful ===
```



Edit with WPS Office