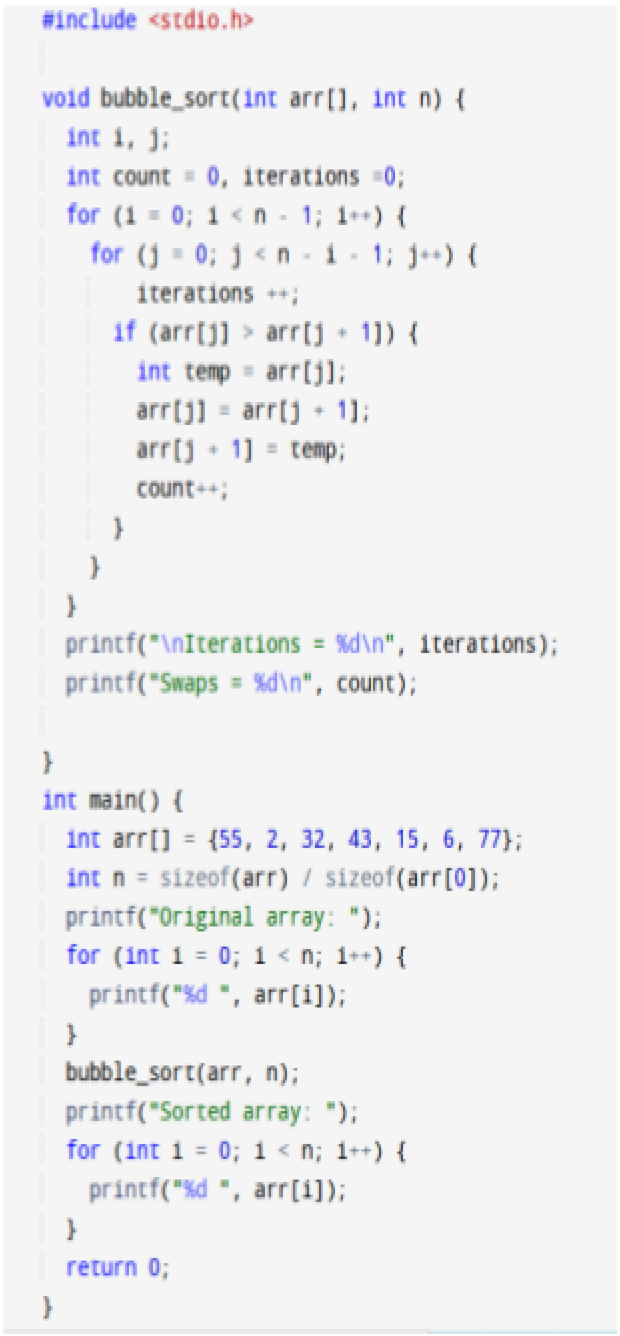
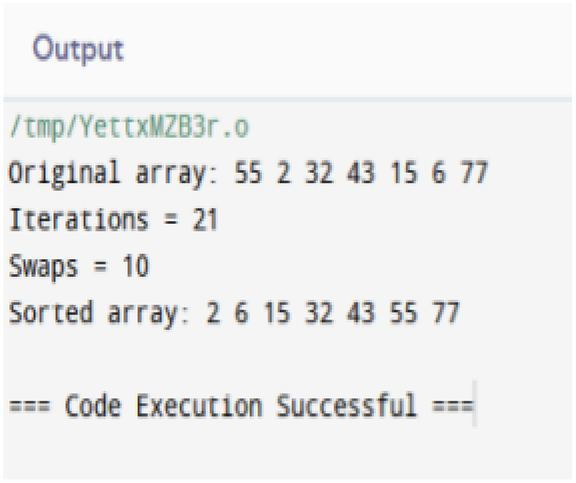
Name: Atharva Salitri **Div: CSAI-B** **Batch: 2** **Roll No.: 37**

Bubble Sort Code and Output:



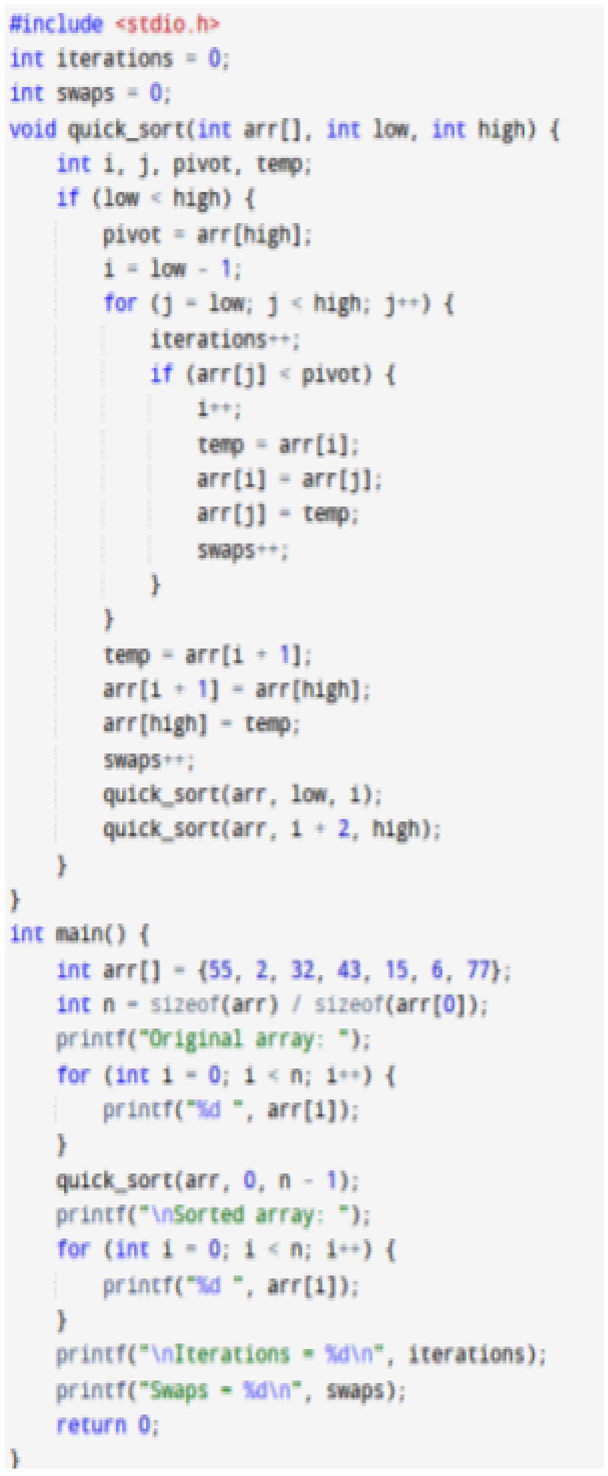
```c
#include <stdio.h>

void bubble_sort(int arr[], int n) {
  int i, j;
  int count = 0, iterations =0;
  for (i = 0; i < n - 1; i++) {
    for (j = 0; j < n - i - 1; j++) {
      iterations ++;
      if (arr[j] > arr[j + 1]) {
        int temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
        count++;
      }
    }
  }
  printf("\nIterations = %d\n", iterations);
  printf("Swaps = %d\n", count);

}
int main() {
  int arr[] = {55, 2, 32, 43, 15, 6, 77};
  int n = sizeof(arr) / sizeof(arr[0]);
  printf("Original array: ");
  for (int i = 0; i < n; i++) {
    printf("%d ", arr[i]);
  }
  bubble_sort(arr, n);
  printf("Sorted array: ");
  for (int i = 0; i < n; i++) {
    printf("%d ", arr[i]);
  }
  return 0;
}
```



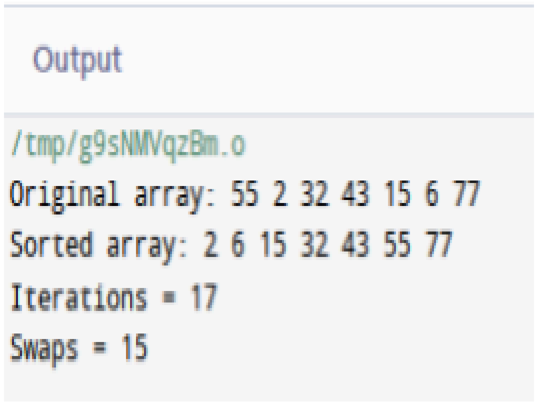
```
/tmp/YettxMZB3r.o
Original array: 55 2 32 43 15 6 77
Iterations = 21
Swaps = 10
Sorted array: 2 6 15 32 43 55 77

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

Name: Atharva Salitri Div: CSAI-B Batch: 2 Roll No.: 37

Quick Sort Code and Output:



```c
#include <stdio.h>
int iterations = 0;
int swaps = 0;
void quick_sort(int arr[], int low, int high) {
    int i, j, pivot, temp;
    if (low < high) {
        pivot = arr[high];
        i = low - 1;
        for (j = low; j < high; j++) {
            iterations++;
            if (arr[j] < pivot) {
                i++;
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
                swaps++;
            }
        }
        temp = arr[i + 1];
        arr[i + 1] = arr[high];
        arr[high] = temp;
        swaps++;
        quick_sort(arr, low, i);
        quick_sort(arr, i + 2, high);
    }
}
int main() {
    int arr[] = {55, 2, 32, 43, 15, 6, 77};
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("Original array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    quick_sort(arr, 0, n - 1);
    printf("\nSorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\nIterations = %d\n", iterations);
    printf("Swaps = %d\n", swaps);
    return 0;
}
```

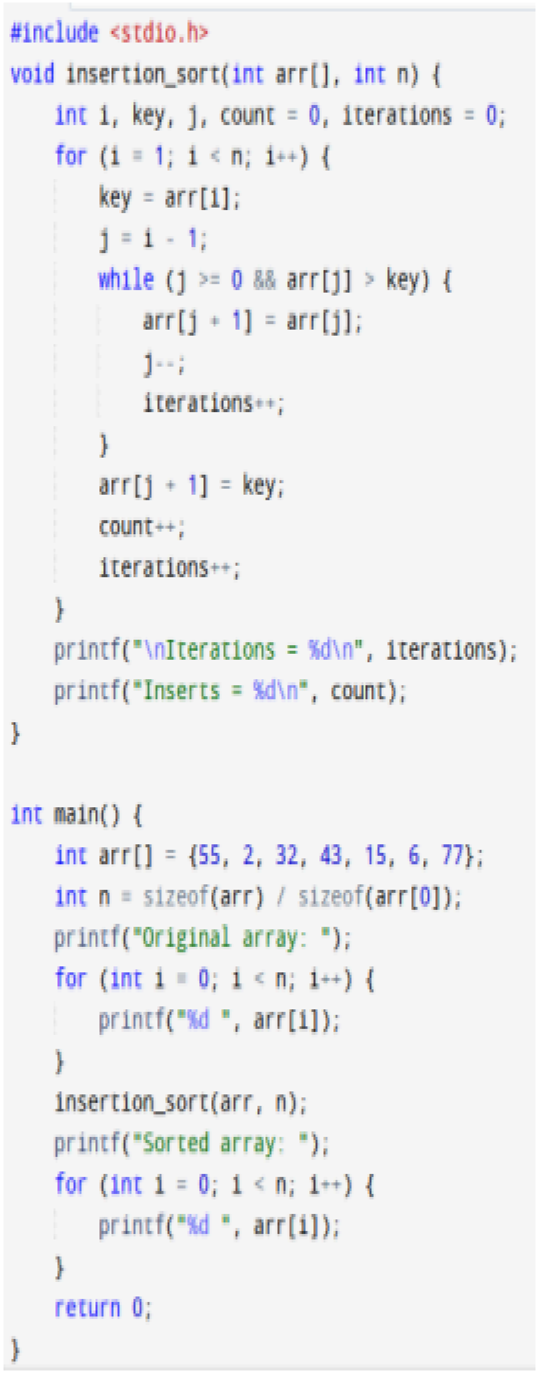


Output

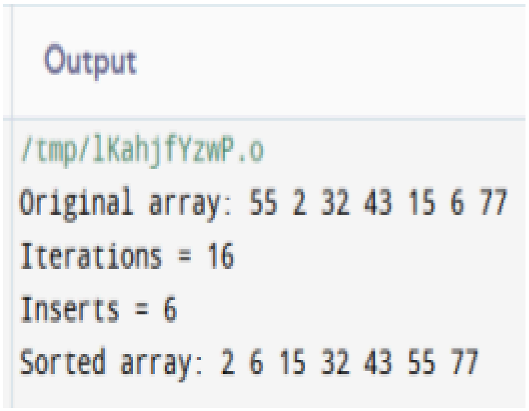
```
/tmp/g9sNMVqzBm.o
Original array: 55 2 32 43 15 6 77
Sorted array: 2 6 15 32 43 55 77
Iterations = 17
Swaps = 15
```

Name: Atharva Salitri **Div: CSAI-B** **Batch: 2** **Roll No.: 37**

Insertion Sort Code and Output:



```c
#include <stdio.h>
void insertion_sort(int arr[], int n) {
    int i, key, j, count = 0, iterations = 0;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
            iterations++;
        }
        arr[j + 1] = key;
        count++;
        iterations++;
    }
    printf("\nIterations = %d\n", iterations);
    printf("Inserts = %d\n", count);
}

int main() {
    int arr[] = {55, 2, 32, 43, 15, 6, 77};
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("Original array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    insertion_sort(arr, n);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```



Output

```
/tmp/lKahjfYzwP.o
Original array: 55 2 32 43 15 6 77
Iterations = 16
Inserts = 6
Sorted array: 2 6 15 32 43 55 77
```