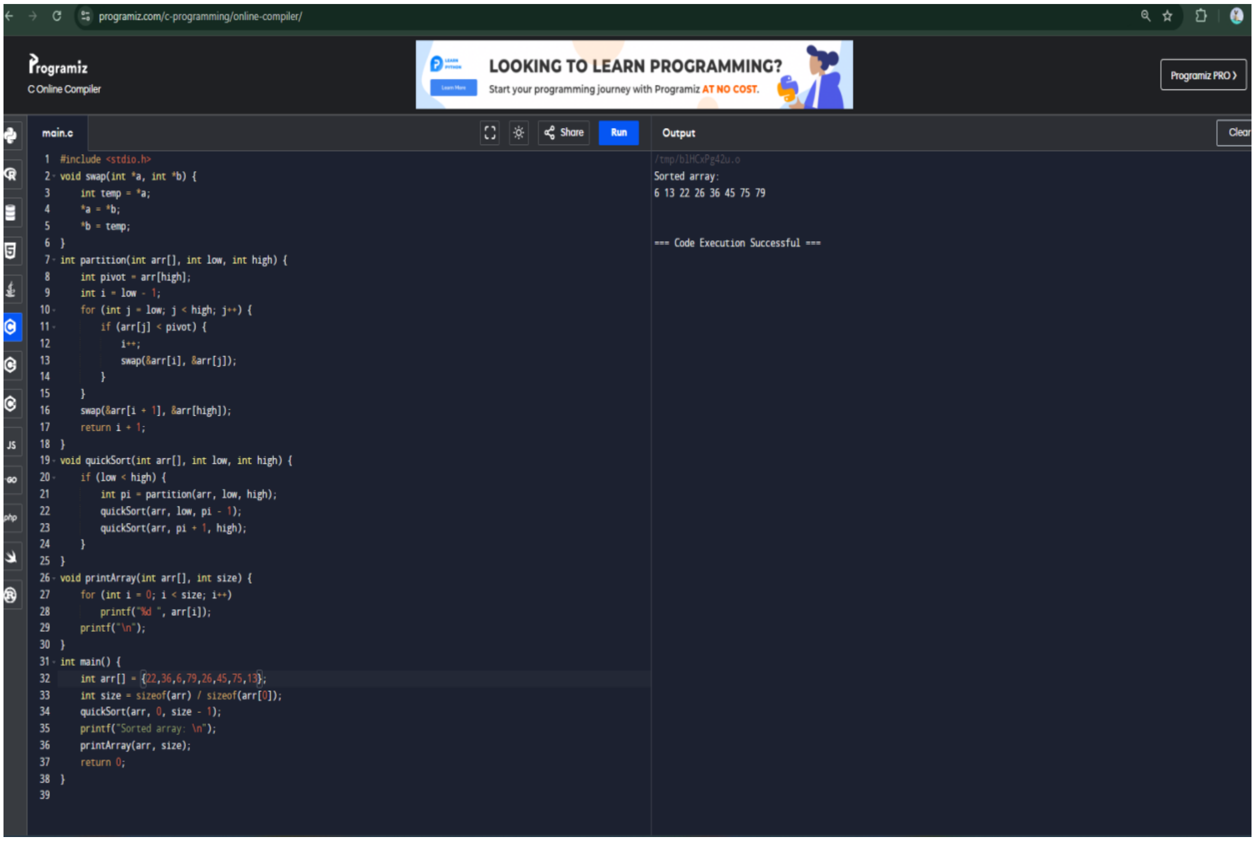
1.Quick Sorting code implementation


programiz.com/c-programming/online-compiler/
Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Programiz PRO ›
main.c
Share
Run
Output
Clear

```c
#include <stdio.h>
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low - 1;
    for (int j = low; j < high; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return i + 1;
}
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
int main() {
    int arr[] = {22,36,6,79,26,45,75,13};
    int size = sizeof(arr) / sizeof(arr[0]);
    quickSort(arr, 0, size - 1);
    printf("Sorted array: \n");
    printArray(arr, size);
    return 0;
}
```

```
/tmp/blHCxPg42u.o
Sorted array:
6 13 22 26 36 45 75 79


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