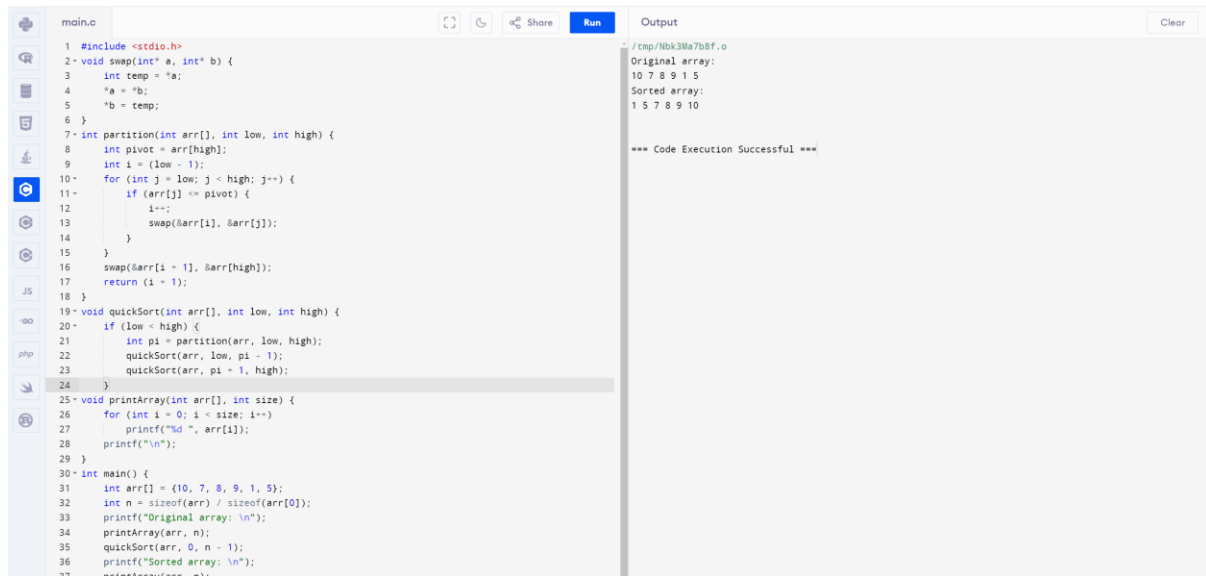


1. Write a c program to evaluate a quick sorting



The image shows a code editor with a C program for Quick Sort. The code is as follows:

```
1 #include <stdio.h>
2 void swap(int* a, int* b) {
3     int temp = *a;
4     *a = *b;
5     *b = temp;
6 }
7 int partition(int arr[], int low, int high) {
8     int pivot = arr[high];
9     int i = (low - 1);
10    for (int j = low; j < high; j++) {
11        if (arr[j] <= pivot) {
12            i++;
13            swap(&arr[i], &arr[j]);
14        }
15    }
16    swap(&arr[i + 1], &arr[high]);
17    return (i + 1);
18 }
19 void quickSort(int arr[], int low, int high) {
20    if (low < high) {
21        int pi = partition(arr, low, high);
22        quickSort(arr, low, pi - 1);
23        quickSort(arr, pi + 1, high);
24    }
25 }
26 void printArray(int arr[], int size) {
27    for (int i = 0; i < size; i++)
28        printf("%d ", arr[i]);
29    printf("\n");
30 }
31 int main() {
32    int arr[] = {10, 7, 8, 9, 1, 5};
33    int n = sizeof(arr) / sizeof(arr[0]);
34    printf("Original array: \n");
35    printArray(arr, n);
36    quickSort(arr, 0, n - 1);
37    printf("Sorted array: \n");
38    printArray(arr, n);
39 }
```

The output of the program is shown on the right:

```
/tmp/Nbk3Ma7b8f.o
Original array:
10 7 8 9 1 5
Sorted array:
1 5 7 8 9 10

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