```
#include <iostream>

using namespace std;

// Function to partition the array and place the pivot element at its correct position

int partition (int arr[], int low, int high)
{
	int pivot = arr[high]; //the pivot
	int i = (low - 1); //the index of smaller element

	for (int j = low; j <= high - 1; j++)
	{
		//if current element is smaller than the pivot
		if (arr[j] < pivot)
		{
			i++; //increment the index of smaller element
			swap(arr[i], arr[j]); //swap the current element with the element at the smaller element index
		}
	}
	swap(arr[i + 1], arr[high]); //swap the pivot element at the smaller element + 1
	return (i + 1); //return the index of the smaller element + 1
}
//this function is recursively sort the array

void quickSort(int arr[], int low, int high)
{
	if (low < high) //if the low index is less than high index
```

```
        {
                int pi = partition(arr, low, high); //partition the array

                quickSort(arr, low, pi - 1); //sort the left subarray

                quickSort(arr, pi + 1, high); //sort the right subarray
        }
}

//the main function of this program.
int main() {

   int arr[] = {10, 6, 7, 8, 9, 4, 5}; //initialize the array with random elements.
   int arrsize = sizeof(arr) / sizeof(arr[0]);

   //print the unsorted array
   cout << "The unsorted array is :  \n";

   for (int i = 0; i < arrsize; i++) {

      cout << arr[i] << " ";
   }
   cout << endl;

   //sort the array using the quick sort algo.
   quickSort(arr, 0, arrsize - 1);

   //print the sorted array
   cout << "The sorted array is : \n";
```

```
    for (int i = 0; i < arrsize; i++) {

        cout << arr[i] << " ";
    }
    cout << endl;

    return 0;
}
```

OUTPUT: