Quick Sort

```
#include<stdio.h>

void swap(int* a, int* b)
{
	int t = *a;
	*a = *b;
	*b = t;
}


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

```
        for (i=0; i < size; i++)
                printf("%d ", arr[i]);
        printf("n");
}

int main()
{
        int arr[] = {10, 7, 8, 9, 1, 5};
        int n = sizeof(arr)/sizeof(arr[0]);
        quickSort(arr, 0, n-1);
        printf("Sorted array: n");
        printArray(arr, n);
        return 0;
}
```