5-Implementation of Quick Sort

Aim:

Write a Program to Implement the Quick Sort Algorithm  
  
Input Format:  
The first line contains the no of elements in the list-n  
The next n lines contain the elements.  
  
Output:  
Sorted list of elements

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  67 34 12 98 78 | 12 34 67 78 98 |

Algorithm:

1. Read the integer n and array arr of size n.
2. Apply the quicksort algorithm to sort the array: choose a pivot, partition the array around the pivot, and recursively sort the subarrays.
3. Print the sorted array.

Code:

#include <stdio.h>

void quicksort(int arr[], int l, int h)

{

if (l < h)

{

int p=arr[h];

int i = l - 1;

for (int j = l; j < h; j++)

{

if (arr[j] < p)

{

i++;

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

int temp = arr[i + 1];

arr[i + 1] = arr[h];

arr[h] = temp;

int mid = i + 1;

quicksort(arr, l, mid - 1);

quicksort(arr, mid+ 1, h);

}

}

int main()

{

int n;

scanf("%d", &n);

int arr[n];

for (int i = 0; i < n; i++)

{

scanf("%d", &arr[i]);

}

quicksort(arr, 0, n - 1);

for (int i = 0; i < n; i++)

{

printf("%d ", arr[i]);

}

}

Output:

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 5  67 34 12 98 78 | 12 34 67 78 98 | 12 34 67 78 98 |  |
|  | 10  1 56 78 90 32 56 11 10 90 114 | 1 10 11 32 56 56 78 90 90 114 | 1 10 11 32 56 56 78 90 90 114 |  |
|  | 12  9 8 7 6 5 4 3 2 1 10 11 90 | 1 2 3 4 5 6 7 8 9 10 11 90 | 1 2 3 4 5 6 7 8 9 10 11 90 |  |

Passed all tests!

**Correct**

Marks for this submission: 1.00/1.00.

Result:

The expected output was obtained