


[Practice](#) [Compete](#) [Jobs](#) [Rank](#) [Leaderboard](#)

 bor0s

Dashboard > Algorithms > Sorting > Insertion Sort - Part 1

Badge Progress
 ★ ★

Points: 651.00 Rank: 47018

Insertion Sort - Part 1

 by HackerRank

[Problem](#)[Submissions](#)[Leaderboard](#)[Discussions](#)

Sorting

One common task for computers is to sort data. For example, people might want to see all their files on a computer sorted by size. Since sorting is a simple problem with many different possible solutions, it is often used to introduce the study of algorithms.

Insertion Sort

These challenges will cover *Insertion Sort*, a simple and intuitive sorting algorithm. We will first start with an already sorted list.

Insert element into sorted list

Given a sorted list with an unsorted number e in the rightmost cell, can you write some simple code to *insert* e into the array so that it remains sorted?

Print the array every time a value is shifted in the array until the array is fully sorted. The goal of this challenge is to follow the correct order of insertion sort.

Guideline: You can copy the value of e to a variable and consider its cell "empty". Since this leaves an extra cell empty on the right, you can shift everything over until V can be inserted. This will create a duplicate of each value, but when you reach the right spot, you can replace it with e .

Input Format

There will be two lines of input:

- **Size** - the size of the array
- **Arr** - the unsorted array of integers

Output Format

On each line, output the entire array every time an item is shifted in it.

Constraints

$$1 \leq \text{Size} \leq 1000$$

$$-10000 \leq e \leq 10000, e \in \text{Arr}$$

Sample Input

```
5
2 4 6 8 3
```

Sample Output

```
2 4 6 8 8
2 4 6 6 8
2 4 4 6 8
2 3 4 6 8
```

Explanation

3 is removed from the end of the array.

In the 1st line $8 > 3$, so **8** is shifted one cell to the right.

In the 2nd line $6 > 3$, so **6** is shifted one cell to the right.

In the 3rd line $4 > 3$, so **4** is shifted one cell to the right.

In the 4th line $2 < 3$, so **3** is placed at position 2.

Task

Complete the method `insertionSort` which takes in one parameter:

- **Arr** - an array with the value e in the right-most cell.

Next Challenge

In the [next Challenge](#), we will complete the insertion sort itself!



Submissions: [79939](#)

Max Score: 30

Difficulty: Easy

Rate This Challenge:

☆☆☆☆☆

[More](#)Current Buffer (saved locally, editable)  

C++



```
18 #include <map>
19 #include <set>
20 #include <list>
21 #include <cmath>
22 #include <ctime>
23 #include <deque>
24 #include <queue>
25 #include <stack>
26 #include <bitset>
27 #include <cstdio>
28 #include <vector>
29 #include <cstdlib>
30 #include <numeric>
31 #include <sstream>
32 #include <iostream>
33 #include <algorithm>
34 using namespace std;
35 void insertionSort(vector<int> ar) {
36     int index = ar.size() - 1;
37     int tmp = ar[index];
38     while (index - 1 >= 0 && ar[index - 1] > tmp) {
39         ar[index] = ar[index - 1];
40
41         for (auto i : ar) cout << i << " "; cout << "\n";
42
43         index--;
44     }
45     ar[index] = tmp;
46     for (auto i : ar) cout << i << " "; cout << "\n";
47 }
48
49 int main(void) {
50     vector<int> _ar;
51     int _ar_size;
52     cin >> _ar_size;
53     for(int _ar_i=0; _ar_i<_ar_size; _ar_i++) {
54         int _ar_tmp;
55         cin >> _ar_tmp;
56         _ar.push_back(_ar_tmp);
57     }
58
59     insertionSort(_ar);
60     return 0;
61 }
62
```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)[Run Code](#)[Submit Code](#)

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)