Intro to Tutorial Challenges

About Tutorial Challenges
Many of the challenges on HackerRank are difficult and assume that you already know the relevant algorithms very well. These tutorial challenges are different. They break down algorithmic concepts into smaller challenges, so that you can learn the algorithm by solving the challenges.

These challenges are intended for those who know some programming and now want to learn some algorithms. You could be a student majoring in Computers, a self-taught programmer, or an experienced developer who wants an active algorithms review!

The first series of challenges covers sorting. The challenges are listed below:

Tutorial Challenges - Sorting

Insertion Sort challenges

- Insertion Sort 1 - Inserting
- Insertion Sort 2 - Sorting
- Correctness and loop invariant
- Running Time of Algorithms

Quicksort challenges

- Quicksort 1 - Partition
- Quicksort 2 - Sorting
- Quicksort In-place (advanced)
- Running time of Quicksort

Counting sort challenges

- Counting Sort 1 - Counting
- Counting Sort 2 - Simple sort
- Counting Sort 3 - Preparing
- Full Counting Sort (advanced)

There will also be some challenges where you'll get to apply what you've learnt.

About the Challenges
The challenges will describe some topic and then ask you to code a solution. As you progress through the challenges, you will learn some important concepts in algorithms. In each challenge, you will receive input on STDIN and you will need to print the correct output to STDOUT.

For many challenges, helper methods (like an array) will be provided for you to process the input into a useful format. You can use these methods to get started with your program, or you can write your own input methods if you want. Your code just needs to print the right output to each test case.

Sample Challenge

This is a simple challenge to get things started. Given a sorted array (ar) and a number (V), can you print the index location of V in the array?

The next section describes the input format. You can often skip it, if you are using included methods.

Input Format
There will be three lines of input:

- V - the value that has to be searched.
- n - the size of the array.
- ar - n numbers that make up the array.

Output Format
Output the index of V in the array.

The next section describes the constraints and ranges of the input. You should check this section to know the range of the input.

Constraints

- $1 \leq n \leq 1000$
- $-1000 \leq V \leq 1000, V \in ar$
- It is guaranteed that V will occur in ar exactly once.

This "sample" shows the first input test case. It is often useful to go through the sample to understand a challenge.

Sample Input

```
4
6
1 4 5 7 9 12
```

Sample Output

```
1
```

Explanation
$V = 4$. The value 4 is the 2^{nd} element in the array, but its index is 1 since array indices start from 0, so the answer is 1.