Practice Exercise #38: Find Pair

http://www.comp.nus.edu.sg/~cs1020/4_misc/practice.html

Objectives:
- Writing efficient algorithm
- Knowing how to analyse algorithm

Task statement:

You are given an integer *K* and an array of integers sorted in ascending order. Find one pair of integers in this array so that the sum of the pair equals *K*.

A class **Pair** is given. You are to use this class in your program **FindPair.java**. You need to submit only **FindPair.java**.

Your program should include a method **findPair()** that returns a pair of distinct indices in the sorted array whose elements sum up to *K*, or null otherwise. If there are more than one pair of elements that sum up to *K*, your method may return any pair.

If there is a solution, your program should print the output as shown in the sample runs below. If there is no solution, your program should print "No solution!"

You may assume that there are at least 2 elements in the array.

Note that in some cases there could be more than one solution. If your algorithm returns a pair of indices that is different from the one returned by our solution, your program will fail CodeCrunch test. In this case, as long as you know that your solution is correct, you may ignore CodeCrunch's result.

You have to ensure that your algorithm runs in **O(*n*)** time, where *n* is the number of elements in the array, instead of O(n^2) time. For this, CodeCrunch won't be able to assess. You have to be able to prove it yourself.

Sample Run #1

```
Enter sum: 26
Enter number of elements: 6
Enter 6 integers: 4 6 8 11 15 19
Values at index 3 (11) and index 4 (15)
```

Sample Run #2

```
Enter sum: 23
Enter number of elements: 8
Enter 8 integers: 1 4 9 15 21 30 35 42
No solution!
```