**Can It Be Sorted?**

Given a list, could you determine if it can be sorted following these rules? You can only perform one of these actions once.

* You can only swap two elements.
* Reverse a sub-sequence in the list.

If the array is already sorted, you will output “yes” and nothing else. If you can sort the array with one of the following rules, then you will output “yes” followed by either “swap” or “reverse” followed by:

* If it’s “swap” then you output the indices of the leftmost value followed by the rightmost value. It will look like this: “yes swap *L R”*.
* If it’s “reverse” then you will output the indices of the leftmost value of the subsequence followed by the rightmost value of the subsequence. It will look like this: “yes reverse *L R*”.

If the list can be sorted both ways, then you will choose “swap” over “reverse”. If the list cannot be sorted, then you will output “no”.

**Input:** The first line of input contains **T**, the number of test cases. Each test case consists of two lines. The first line contains **N**, the number of numbers on the following line. The next line contains **N** space-separated integers. Each integer will be distinct.

**Output:** Print the results of each test case like this: “CASE #1: (answer)”.

**Example Input:**

3

2

4 2

3

3 1 2

6

1 5 4 3 2 6

**Example Output:**

CASE #1: yes swap 1 2

CASE #2: no

CASE #3: yes reverse 2 5

**Explanation:** In case #1, you can swap the integers 4 and 2 around, so you will output the index of the 4, followed by the index of the 2.

In case #2, any possible swap you could make would still result in the list being unsorted. The reversal of and sub-sequence of the list would also result in it being unsorted. So, we output “no”.

In case #3, you cannot perform and swaps, but you can perform a reversal to sort the list. By reversing the sub-sequence from indices [2…5] (which are the numbers 5, 4, 3, 2), you can successfully sort the list. So, in this case, we print “yes reverse 2 5”.