

2 King's Palindrome List

2.1 General information

Your submission for all the parts of the exercise should be in Java. All the challenge exercises must be done in pairs.

2.2 Assignment description

The Kingdom of Palindrome numbers has designated Peter the Programmer to solve a problem of the King's Palindrome List: the advisor of the king gave Peter a list containing N natural numbers, each with an odd number of digits. A number is a palindrome if its first digit is equal to the last, the second is equal to the penultimate, and so on. He told Peter that the king is very old and ill. As a result, the numbers on his list became sometimes wrong. The advisor told Peter that the correct list can be obtained by replacing each number in it with the smallest palindrome greater than or equal to that number.

Following the advice of the counselor, Peter the Programmer restored the list and found out that in the correct list all palindromes appeared to be distinct. Looking more closely at the palindromes on this list, he noted that there are pairs of palindromes in which the smallest can be obtained from the biggest by removing from it the same number of digits on both ends. For example, for the pair 7531357 and 313 the latter can be obtained from the former by removing two digits from both ends of it.

We consider a subset of palindromes from the correct list and denote as X the maximum value in this subset. We will say that such a subset is a magic set if all the palindromes in it can be obtained from X by the method described above. An example of a magic set is 4, 53435, 7534357, 89753435798, assuming that all these numbers, among maybe other numbers, are present in the correct list.

2.3 Exercise

Write a program that reads the numbers from the list received from the advisor, and can display the output of each one of the tasks below:

- Task 1: The correct list obtained by Peter the Programmer;
- Task 2: The number of elements of the largest magic set that can be obtained from the correct list.
- Task 3: The palindromes from which the largest magic set is formed, displayed in ascending order. If there are several such largest magic sets on the correct list, the one with the largest X is displayed. If there are no magic sets, the largest number in the correct list is displayed.

2.4 Input

The input contains on the first line one of the natural numbers 1, 2 or 3 that indicates the number of the task that needs to be solved in this program run. The second line contains the natural number N which represents the number of elements in the list received from the king's advisor. The third line contains the natural numbers from the corrupt list received from the advisor, separated by a space.

2.5 Output

The output should be one line that contains the output of the requested task. If the first task has been solved, the output will contain N natural numbers, separated by a space, representing the numbers in the correct list in the order corresponding to the initial list. If task 2 is solved, the size of the largest magic set should be printed. If task 3 is solved, the output should either contain the elements of the largest magic set (with the largest X) separated by a space, or the largest number in the correct list, in case no magic sets exist.

2.6 Examples

Input	Output	Explanation
1 3 345 214 64325	353 222 64346	The correct palindrome list.
2 6 2 3 4 432 5643461 7	3	The correct list contains the palindromes: 2 3 4 434 5643465 7 and the largest magic set is 3 434 5643465.
3 5 489 123 14890 2148909 789	494 14941 2149412	The correct list contains the palindromes: 494 131 14941 2149412 797.
3 4 123 456 789 1053432	1053501	Largest element (no magic sets found).

2.7 Specifications

1. $0 < N \leq 50000$;
2. The numbers on the input list are **natural** (positive \mathbb{Z}^+ integers) and each has a maximum of **17 digits**;

3. The input is assumed to be in the right format and you don't need to check it. In other words, the program doesn't need to be robust.
4. As your program is going to be automatically checked by Momotor, please, avoid in the output any (except an empty line in the end) extra strings or any extra characters, including leading or trailing spaces.

2.8 Minimal test set

Before you start programming, create a *minimal test set* in the file `minimal_test_set.txt`. Add at least six different test cases, two for each task that the program should be able to perform. Only add test cases that are complementary to the examples given above. For instance, a case that addresses task 3 in a situation where no magic set can be found would be superfluous. Also, don't add multiple test cases that test the same behaviour.

Motivate each test case. For each test case, give the *exact* input and the *exact* output you expect.

2.9 Manual

Once you've created the minimal test set, you should have a good understanding of the program's behavior. Use that understanding to write a short, but complete user manual for the `KingsPalindromeList` program. Use your own words. Write the words in the given Java template file, starting at `TODO: Documentation`.

2.10 Restrictions and recommendations

To make the exercise more challenging, you are discouraged to use strings (except in the output) and `ArrayList` types. Violations will downgrade your mark up to 3 points. Instead, you can use arrays or/and `Array` types. Using methods of the `Math` class is allowed.