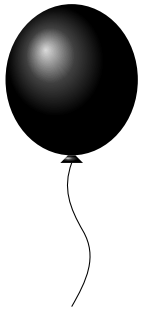


G Shuffling the Deck

TIME LIMIT: 2.0s
MEMORY LIMIT: 512MB



Automation are rapidly invading all areas of our lives. Dishwashers, onboard computers in cars, electric fly swatters — all these devices to some extent make standard familiar actions easier for humans. In this problem, you will be required to automate yet another familiar action for many people — shuffling a deck of cards.

Let's consider a new deck of cards, where the cards have a certain characteristic that is represented by an integer from 1 to t . We will call it the value of the card, and several cards in the deck can have the same value.

Initially, the cards in the deck are sorted in non-decreasing order of their values. Accordingly, the first card in the deck has a value of 1, and the last card has a value of t . Moreover, the values of any two consecutive cards in the deck differ by at most 1, and the value of each subsequent card is not less than the value of the previous card.

We will call a deck *well-shuffled* if no two consecutive cards have the same value.

To shuffle the deck a device is used that can perform actions of the following type: for some i and j , take the cards that are in positions from i to j , inclusive, and move them to the end of the deck in the same order.

You need to write a program that, given the description of the initial deck, will create a sequence of the minimum number of specified actions, the execution of which will result in a well-shuffled deck.

INPUT

The first line contains a single positive integer x — the number of decks that need to be shuffled. Then follows the description of x decks.

The description of each deck consists of two lines. The first line of the deck description contains a positive integer n — the number of cards in the deck. The next line contains n integers that describe the values of the cards in the deck. The first number in such a line is always equal to 1, and each subsequent number is not less than the previous one and differs from it by at most one.

The total number of cards in all decks in the input does not exceed 1 000 000.

OUTPUT

For each deck given in the input, print -1 if it is impossible to shuffle it using the actions described in the problem statement. If the shuffling algorithm exists, then in the first line of its description, print k — the minimum number of actions required to shuffle the deck. In the next k lines, print the actions, each of which is described by two integers i and j — the positions of the cards that define the consecutive subsequence that needs to be moved to the end.

SAMPLES

Sample input 1	Sample output 1
2 8 1 1 2 2 3 3 4 4 6 1 1 1 1 2 3	2 2 3 4 5 -1