# 1165 – Digit Dancing

Digits like to dance. One day, 1, 2, 3, 4, 5, 6, 7 and 8 stand in a line to have a wonderful party. Each time, a male digit can ask a female digit to dance with him, or a female digit can ask a male digit to dance with her, as long as their sum is a prime. Before every dance, exactly one digit goes to who he/she wants to dance with - either to its immediate left or immediate right.

For simplicity, we denote a male digit x by itself x, and denote a female digit x by -x. Suppose the digits are in order {1, 2, 4, 5, 6, -7, -3, 8}. If -3 wants to dance with 4, she must go either to 4's left, resulting {1, 2, -3, 4, 5, 6, -7, 8} or his right, resulting {1, 2, 4, -3, 5, 6, -7, 8}.

Note that -3 cannot dance with 5, since their sum 3+5 = 8 is not a prime; 2 cannot dance with 5, since they're both male.

Given the initial ordering of the digits, find the minimal number of dances needed for them to sort in increasing order (ignoring signs of course).

## Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case contains exactly eight integers in a single line. The absolute values of these integers form a permutation of **{1, 2, 3, 4, 5, 6, 7, 8}**.

## Output

For each test case, print the case number and the minimal number of dances needed. If they can never be sorted in increasing order, print **-1**.

| Sample Input | Output for Sample Input |
|---|---|
| 5<br>1 2 4 5 6 -7 -3 8<br>1 2 3 4 5 6 7 8<br>1 2 3 5 -4 6 7 8<br>1 2 3 5 4 6 7 8<br>2 -8 -4 5 6 7 3 -1 | Case 1: 1<br>Case 2: 0<br>Case 3: 1<br>Case 4: -1<br>Case 5: 3 |