

Spring Primality

Dickie was playing with his friend some day in the spring season, he bought some cards, each card contains a number, that can be a prime or not. The game that Dickie plays consists to count what is the longest segment of contiguous cards that contains a prime, by instance, if the cards are: 2 5 8 14 11 15, the longest contiguous segment will be of length 2 as the cards 2,5 are primes, while the 11 is automatically discarded as it is isolated.

Nevertheless, Dickie's friends are evil and they challenged him to do another task, that goes as follows, for every card that is not a prime, you subtract every of its prime factors to the prime count of the factor and for every card that is a prime, you will add a unit to its prime count, i.e.: 2 5 8 14 11 15, primes=2,5,11, $8=2*2*2$, $14=2*7$, $15=3*5$, so, prime count [2] = 1- 3-1, prime count [5] = 1-1 prime count [11] = 1. There is no consideration for the other numbers.

After doing this, Dickie will count another time the longest contiguous segment, however, if the prime count is less or equal than zero, you can't count that prime card. Help Dickie to find the longest contiguous segment of cards according to his rules, and the longest contiguous segment of cards according to his friends rules.

Input

The first line contains an integer T , which specifies the number of test cases. Then, will follow the descriptions of T test cases.

The first line of each case will contain a number N , then, in the next line, N numbers will follow.

Output

For each test case you should print the string "Scenario #i: " where i represents the test case you are analyzing (starting from 1), followed by the longest segment of contiguous prime cards, then, a single space, a "greater than-sign" and another single space will follows and after that, the segment of contiguous cards according to the rules proposed by Dickie's friends.

INPUT	OUTPUT
3	Scenario #1: 4 > 2
7	Scenario #2: 2 > 1
7 9 11 2 5 7 6	Scenario #3: 1 > 1
3	
2 3 4	
5	
8 16 32 64 7	

Constraints - Subtask 1 (20%)

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- $2 \leq N_i \leq 1000$

Constraints - Subtask 2 (30%)

- $1 \leq T \leq 10$
- $1 \leq N \leq 1000$
- $2 \leq N_i \leq 100000$

Constraints - Subtask 3 (50%)

- $1 \leq T \leq 5$
- $1 \leq N \leq 100000$
- $2 \leq N_i \leq 10000000$