Fast Food

Lea wants to meet her friend Bea at a new restaurant near the campus of the Thomas Underwood University (TUM). Since most of the customers are math students, the owner decided to print the place cards as follows: Each card contains the row a and column b of the table (tables are ordered in a perfect grid) as well as the greatest common divisor, largest common multiple and of course the prime factors of a and b.

Since simply telling your friend at which table you sit would be boring, some customers have invented the following fun game: Instead of telling your friend the exact table-coordinates, you send the gcd of the table you are sitting at as well as the gcds of a few tables to the right. I.e. if you are sitting at table (a, b), you will send $gcd(a, b), gcd(a + 1, b), \ldots, gcd(a + k, b)$ for some k. Your friend then has to figure out at which table you might sit. Because there might be multiple solutions, it has become a custom to tell the friend the right coordinates once he has come up with one correct solution.

Because most of the students communication is performed using Twottr, a short messaging service which allows only 39 letters per message, a and b can each have at most 19 digits.

Of course Bea takes part in this game and has sent Lea some gcds. Can you help Lea figure out where Bea might sit?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer k, the number of gcds. The next line contains k numbers $a_0, \ldots, a_{k-1}$, where $gcd(a, b) = a_0$, $gcd(a + 1, b) = a_1, \ldots, gcd(a + k - 1, b) = a_{k-1}$.

Output

For each test case, output one line containing "Case #i: a b" where i is its number, starting at 1, and a, b are integers s.t. $gcd(a, b) = a_0$, $gcd(a + 1, b) = a_1, \ldots, gcd(a + k - 1, b) = a_{k-1}$ and $1 \leq a, b \leq 10^{19}$.

Constraints

- $1 \leq t \leq 20$
- $3 \leq k \leq 1000$
- $1 \leq a_i \leq 10^{19} - k$

Sample Input 1

```
2
4
3 2 1 6

4
1 2 3 4
```

Sample Output 1

```
Case #1: 3 6
Case #2: 1 12
```

Sample Input 2

```
9
3
5 4 3

5
4 5 6 7 8

10
10 3 2 1 6 5 2 3 2 11

7
3 2 1 30 17 2 3

9
7 2 3 2 5 6 1 14 9

10
3 14 11 12 1 2 3 4 7 6

6
84 1 2 3 4 11

6
2 3 14 5 6 1

4
18 7 10 3
```

Sample Output 2

```
Case #1: 55 60
Case #2: 4 840
Case #3: 200 330
Case #4: 387 510
Case #5: 91 630
Case #6: 405 924
Case #7: 336 924
Case #8: 152 210
Case #9: 468 630
```