Candies

Most people are quite happy to invite friends. Lea is too and of course she strives to make the invitees as happy as possible. This sometimes proves to be quite difficult. This time she plans to buy candies for everyone. While this sounds like a simple task, the eating habits of her friends complicate things.

When eating candies, all friends sit in a big circle. All Candies are poured into a big bowl that is passed around. Each friend has a specific number of candies that he eats every time he gets the bowl. The bowl starts full at Lea, is passed around each time in the same order, and Lea always eats exactly one candy whenever she gets the bowl back.

She now wants to buy a number of candies such that, no matter which of her friends show up, the bowl will end up empty after Lea takes a candy (Then the bowl is passed around no more). It may be passed around a couple of times, but it should not happen that a friend cannot take his number of candies or that it returns to Lea empty.

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 n, the number of Lea's friends. The next line contains a space separated list of n integers $c_1 ... c_n$, c_i is the number of candies her i-th friend eats each time he gets the bowl.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the minimal number of candies Lea has to buy to satisfy the constraints above. You can assume that at least one friend will show up. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 15$
- $1 \leq c_i \leq 10$
- The number x will be at most $2^{63} - 1$

Sample Input 1	Sample Output 1
2 2 3 5 3 1 1 1	Case #1: 36 Case #2: 12

Sample Input 2

```
10
1
4

3
4 2 4

2
3 2

3
3 1 4

3
2 1 4

2
3 3

2
3 2

3
1 2 1

3
2 2 3

3
1 2 2
```

Sample Output 2

```
Case #1: 5
Case #2: 3465
Case #3: 12
Case #4: 360
Case #5: 840
Case #6: 28
Case #7: 12
Case #8: 60
Case #9: 120
Case #10: 60
```