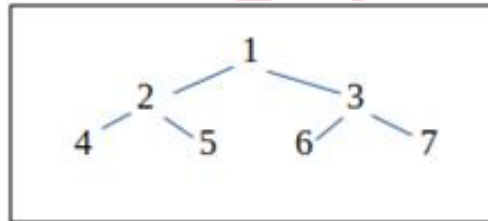


Sum of The Leaves

Poki gets an assignment from his teacher to count the number of leaves from a tree in the following way.



The numbers describe the number of leaves. Poki's job is to count from top to bottom in the direction of the left or right branches. In the picture above, the number of leaves from top to bottom branch is 7, 8, 10, 11.

- 7 is obtained from $1 + 2 + 4$
- 8 is obtained from $1 + 2 + 5$
- 10 is obtained from $1 + 3 + 6$
- 11 is obtained from $1 + 3 + 7$

Poki asks for your help to create a program that can help count the number of leaves.

Format Input

The input consists of one integer T , the number of test cases given. For each test case, there is a number N in the next line followed by a series of N numbers A_i .

Format Output

The output must be declared in the "Case #X:\n" format, where X is the test case number starting at 1, and then the next line represents number of D leaves per lowest branch. The lowest branch is a branch that has no other branches below it.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- $1 \leq A_i \leq 100$

Sample Input (standard input)

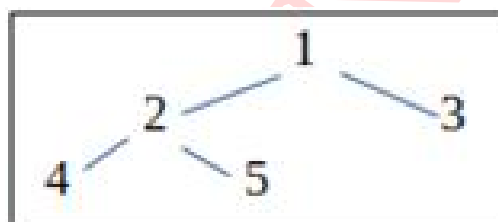
```
3
7
1 2 3 4 5 6 7
5
1 2 3 4 5
2
3 3
```

Sample Output (standard output)

```
Case #1:
7
8
10
11
Case #2:
7
8
4
Case #3:
6
```

Explanation for Case 2

The order of the inputs to form a tree is from left to right then down with a maximum of two branches.



- 7 is obtained from $1 + 2 + 4$
- 8 is obtained from $1 + 2 + 5$
- 4 is obtained from $1 + 3$

In the illustration above, the array used can be described as follows:

A[1]	A[2]	A[3]	A[4]	A[5]
1	2	3	4	5

To get the left branch you can use the formula $2 \times \text{index}$. To get the right branch you can use the formula $(2 \times \text{index}) + 1$.

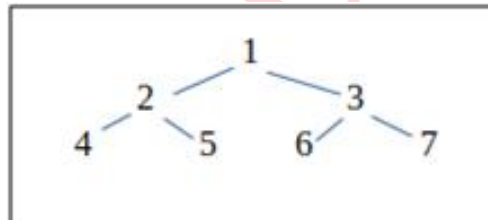
Example:

- left and right branches of A[1] are A[2] and A[3]
- left and right branches of A[2] are A[4] and A[5]
- left and right branches of A[3] do not exist. However, if there were branches of A[3] then it would be placed into A[6] and A[7]

Do the process recursively to count the number of leaves in the tree.

Sum of The Leaves

Poki mendapatkan tugas dari gurunya untuk menghitung jumlah daun dari sebuah pohon dengan cara sebagai berikut.



Angka menggambarkan jumlah daun. Tugas Poki adalah menghitung dari atas ke bawah sesuai dengan arah cabang kiri atau kanan. Pada gambar di atas maka jumlah daun dari atas hingga cabang terbawah adalah 7, 8, 10, 11.

- 7 didapat dari $1 + 2 + 4$
- 8 didapat dari $1 + 2 + 5$
- 10 didapat dari $1 + 3 + 6$
- 11 didapat dari $1 + 3 + 7$

Poki meminta bantuan anda untuk membuat program yang dapat membantu menghitung jumlah daun.

Format Input

Input terdiri dari satu bilangan bulat T , yaitu jumlah test case yang diberikan. Untuk setiap test case, ada sebuah bilangan N di baris berikutnya kemudian diikuti dengan sederet angka A_i sebanyak N buah.

Format Output

Output harus dinyatakan dalam format “Case # X : $\backslash n$ ”, di mana X adalah nomor test case mulai dari 1, yang kemudian baris berikutnya merupakan jumlah daun D per cabang terbawah. Cabang terbawah adalah cabang yang sama sekali tidak memiliki cabang lain di bawahnya.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- $1 \leq A_i \leq 100$

Sample Input (standard input)

```
3
7
1 2 3 4 5 6 7
5
1 2 3 4 5
2
3 3
```

Sample Output (standard output)

Case #1:

```
7
8
10
11
```

Case #2:

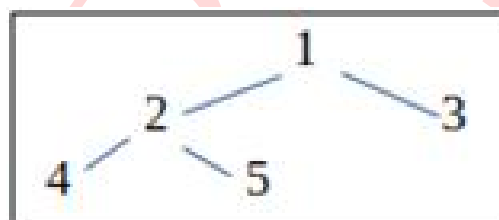
```
7
8
4
```

Case #3:

```
6
```

Penjelasan untuk Case 2

Urutan input hingga berbentuk pohon adalah dari kiri ke kanan kemudian ke bawah dengan jumlah cabang maksimal dua.



- 7 didapat dari $1 + 2 + 4$
- 8 didapat dari $1 + 2 + 5$
- 4 didapat dari $1 + 3$

Pada ilustrasi di atas, array yang digunakan bisa digambarkan sebagai berikut:

A[1]	A[2]	A[3]	A[4]	A[5]
1	2	3	4	5

Untuk mendapatkan cabang kiri bisa menggunakan rumus $2 \times \text{indeks}$. Untuk mendapatkan cabang kanan bisa menggunakan rumus $(2 \times \text{indeks}) + 1$.

Misalnya:

- cabang kiri dan kanan dari A[1] adalah A[2] dan A[3]
- cabang kiri dan kanan dari A[2] adalah A[4] dan A[5]
- cabang kiri dan kanan dari A[3] tidak ada. Namun jika ada, maka akan ditempatkan ke dalam A[6] dan A[7]

Lakukan proses secara rekursif untuk menghitung jumlah daun pada pohon tersebut.