

Problem A: Iterated Difference

Source: `diff.{c,cpp,java}`

Input: `console {stdin,cin,System.in}`

Output: `console {stdout,cout,System.out}`

You are given a list of N non-negative integers $a(1), a(2), \dots, a(N)$. You replace the given list by a new list: the k -th entry of the new list is the absolute value of $a(k) - a(k+1)$, wrapping around at the end of the list (the k -th entry of the new list is the absolute value of $a(N) - a(1)$). How many iterations of this replacement are needed to arrive at a list in which every entry is the same integer?

For example, let $N = 4$ and start with the list (0 2 5 11). The successive iterations are:

```

2 3 6 11
1 3 5 9
2 2 4 8
0 2 4 6
2 2 2 6
0 0 4 4
0 4 0 4
4 4 4 4

```

Thus, 8 iterations are needed in this example.

Input

The input will contain data for a number of test cases. For each case, there will be two lines of input. The first line will contain the integer N ($2 \leq N \leq 20$), the number of entries in the list. The second line will contain the list of integers, separated by one blank space. End of input will be indicated by $N = 0$.

Output

For each case, there will be one line of output, specifying the case number and the number of iterations, in the format shown in the sample output. If the list does not attain the desired form after 1000 iterations, print 'not attained'.

Sample Input

```
4
0 2 5 11
5
0 2 5 11 3
4
300 8600 9000 4000
16
12 20 3 7 8 10 44 50 12 200 300 7 8 10 44 50
3
1 1 1
4
0 4 0 4
0
```

Sample Output

```
Case 1: 8 iterations
Case 2: not attained
Case 3: 3 iterations
Case 4: 50 iterations
Case 5: 0 iterations
Case 6: 1 iterations
```