

# Escalator

By Crithian Bonilha, UTFPR  Brazil

**Timelimit: 1**

Escalators have facilitated people's lives a lot. Climbing stairs is one of the most boring tasks ever invented (after the invention of the stairs on the first place).

After a few observations, you realized that there is a lot of energy being spent with escalators, because they continue to work even when there is nobody using it. To avoid that, the owner of a local shopping mall installed a sensor that verifies when there is someone on the escalator. When the sensor detects no presence, the escalator is deactivated, this way saving energy until the next person arrives.

To be more precise, the system works on the following way: the escalator is initially inactive. The amount of time that a person takes to go from the beginning to the end of the escalator is 10 seconds. In other words, if one person arrives to the escalator on time  $t$ , the escalator will be active on times  $t, t+1, t+2, \dots, t+8$  and  $t+9$ , and then will be deactivated on time  $t+10$ , when the person leaves the escalator. Such time window may be prolonged if one or more people arrive to the escalator during such process.

The owner of the local shopping mall now asked for your help. Write an algorithm that, given the times at which some people arrived at the escalator, tell for how many seconds the escalator was active.

## Input

There will be at most 30 test cases. Each test case starts with a line containing one integer  $N$ , representing the number of people that used the escalator on that day ( $1 \leq N \leq 100$ ).

On the next line, there will be  $N$  distinct integers, given in ascending order, representing the time  $t$  at which each person arrived at the escalator ( $1 \leq t \leq 1000$ ).

The last test case is indicated when  $N = 0$ , which should not be processed.

## Output

For each test case print one line, containing one integer, representing how many seconds the escalator was active.

Input Sample	Output Sample
1	10
5	20
2	13
12 25	36
2	
13 16	

Input Sample	Output Sample
5	
15 20 29 31 50	
0	