# 7950   Toys and Triangles

Alaa fondly remembers playing with a building toy when she was a child. It consisted of segments that could be fastened at each end. A game she liked to play was to start with one segment as a base, placed flat against a straight wall. Then she repeatedly added on triangles, with one edge of the next triangle being a single segment already in place on her structure, and the other two sides of the triangle being newly added segments. Of course no segment could go through the wall, but she did allow newly added segments to cross over already placed ones. Her aim was to see how far out from the wall she could make her structure go.

Distance: 42.3374133

Distance: 29.4619634
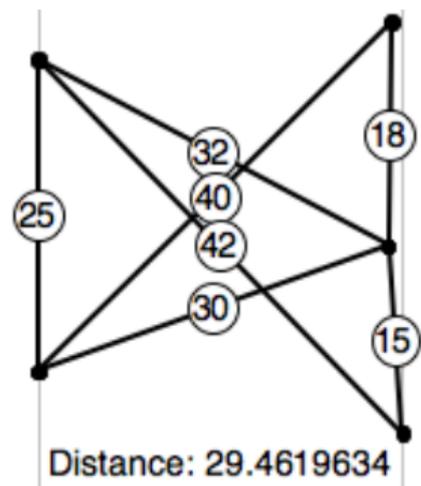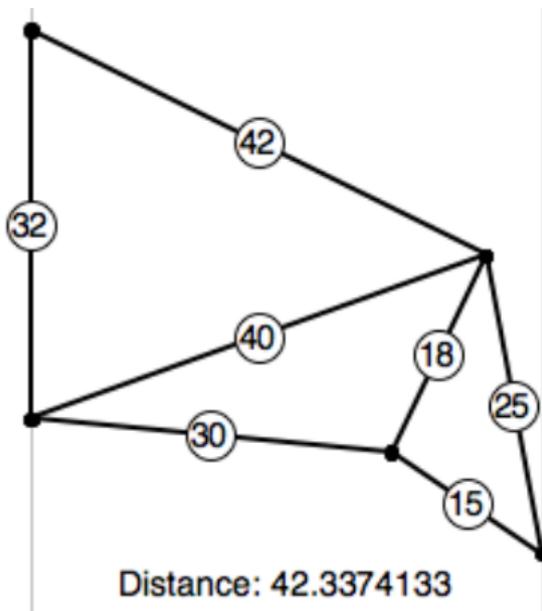
She would experiment, building different ways with different combinations of some or all of her pieces. It was an easy, boring task if all the segments that she used were the same length! It got more interesting if she went to the opposite extreme and started from a group of segments that were all of distinct lengths.

Distance: 66.9495287
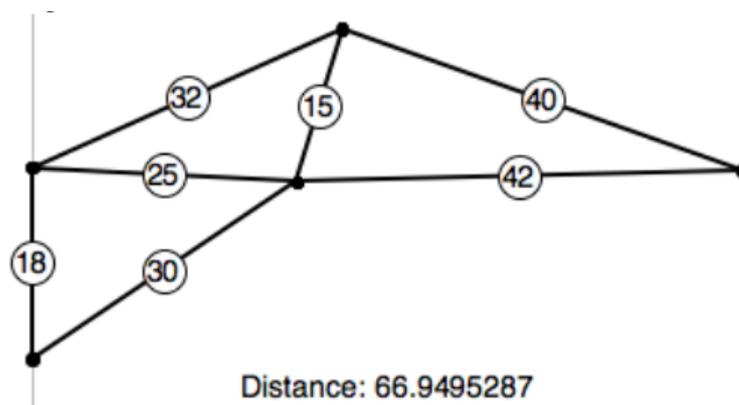
For instance, with segments of length 42, 40, 32, 30, 25, 18 and 15, the figures above show some of the structures she could have built, including the one with the maximum distance.

Now looking back as a Computer Science student, Alaa wondered how well she did, and decided to write a program to check what the maximum distance really is. This is also the challenge for you.

Input

The input file contains several test cases, each of them is a single line of positive integers,

$n\ L_1\ L_2\ \ldots\ L_n$

Here n is the number of segments, $3 \le n \le 10$, and then the lengths of the segments are listed in strictly decreasing order. Each segment length is less than 100. At least one triangle may be constructed.

Output

For each test case, print on a line by itself, the maximum distance that one of Alaa's structures can go from the wall to two decimal digits of precision.

Sample Input

```
3 50 40 30
4 50 40 30 29
7 42 40 32 30 25 18 15
```

Sample Output

```
40.00
40.00
66.95
```