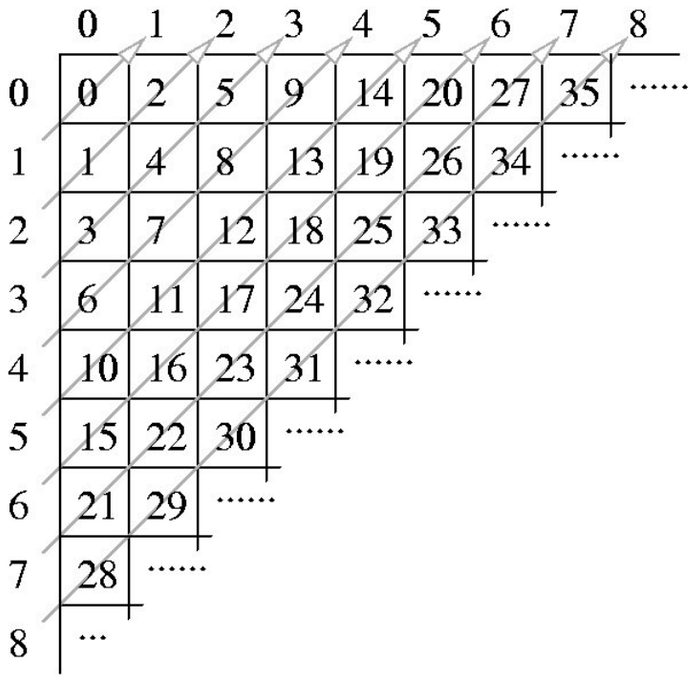
Imagine an infinite table with rows and columns numbered using the natural numbers. The following figure shows a procedure to traverse such a table assigning a consecutive natural number to each table cell:



This enumeration of cells can be used to represent complex data types using natural numbers:

- A pair of natural numbers (i, j) is represented by the number corresponding to the cell in row i and column j. For instance, the pair (3,2) is represented by the natural number 17; this fact is noted by $P_2(3, 2) = 17$.
- The pair representation can be used to represent n-tuples. A triplet (a, b, c) is represented by $P_2(a, P_2(b, c))$. A 4-tuple (a, b, c, d) is represented by $P_2(a, P_2(b, P_2(c, d)))$. This procedure can be generalized for an arbitrary n:

  $$P_n(a_1, ..., a_n) = P_2(a_1, P_{n-1}(a_2, .., a_n)),$$

  where P_n denotes the n-tuple representation function, $n \geq 2$. For example $P_3(2, 0, 1) = 12$.
- A list of arbitrary length $\langle a_1, ..., a_n \rangle$ is represented by

  $$L(\langle a_1, ..., a_n \rangle) = P_2(n, P_n(a_1, ..., a_n)).$$

  For example, $L(\langle 0, 1 \rangle) = 12$.

The Association of Convex Makers (ACM) uses this clever enumeration scheme in a polygon representation system. The system can represent a polygon, defined by integer coordinates, using a natural number as follows: given a polygon defined by a vertex sequence $\langle (x_1, y_1), ..., (x_n, y_n) \rangle$ assign the natural number:

$$L(\langle P_2(x_1, y_1), ..., P_2(x_n, y_n) \rangle).$$

ACM needs a program that, given a natural numbers that represents a polygon, calculates the area of the polygon. It is guaranteed that the given polygon is a simple one, i.e. its sides do not intersect.

As an example of the problem, the triangle with vertices at (1,1), (2,0) and (0,0) is codified with the number 2141. The area of this triangle is 1.

Input

The input consists of several test cases. Each test case is given in a single line of the input by a natural number representing a polygon. The end of the test cases is indicated with '*'.

Output

One line per test case, preserving the input order. Each output line contains a decimal number telling the area of the corresponding encoded polygon. Areas must be printed with 1 decimal place, truncating less significative decimal places.

Sample Input

```
2141
206
157895330
*
```

Sample Output

```
1.0
0.5
1.0
```