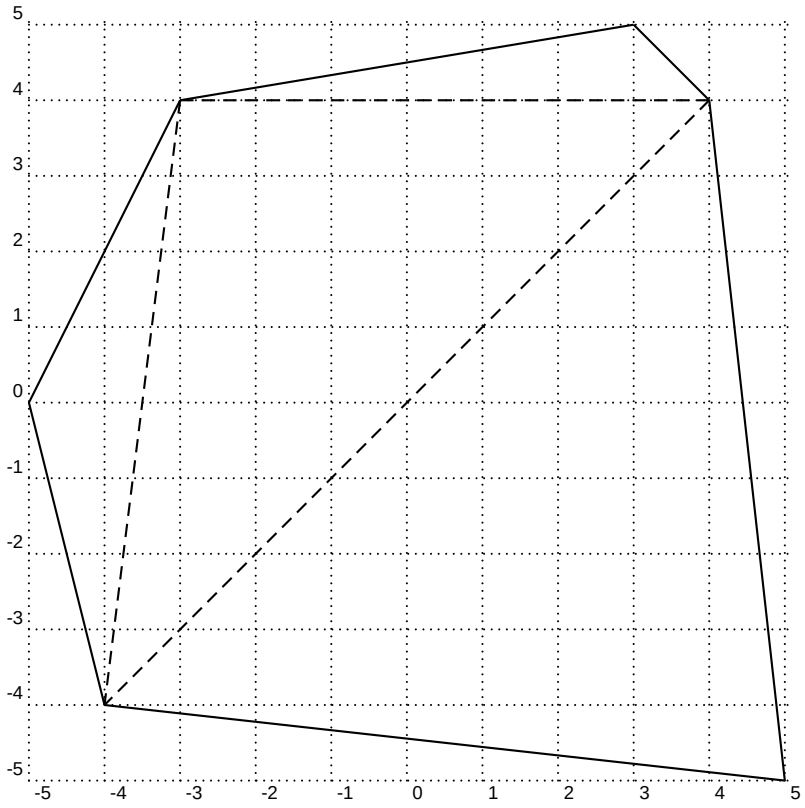
Optimal Triangulation

Time Limit: 2 seconds

Problem Description

A polygon is convex if every internal angle is less than π. A triangulation of a polygon is decomposing it into triangles by line segments. For example, the solid lines in the following figure form a convex polygon, and the dashed line segments triangulate this convex polygon.



Given a convex polygon $(p_1, \ldots, p_n)$, find the optimal triangulation that minimizes the total length of the line segments.

Technical Specifications

1. The number of test cases is no more than 20.
2. Basic input: $4 \leq n \leq 20$
3. Hard input: $4 \leq n \leq 200$
4. $p = (x, y) \in [-1000, 1000] \times [-1000, 1000]$ for every vertex p of the polygon.

Input Format

The first line of the input file contains an integer indicating the number of test cases. The first line of each test case contains an integer n which denotes the number of vertices of the polygon. The following n lines represent the vertices $p_1, \ldots, p_n$ in the clockwise order.

Output Format

For each test case, output the total length of the line segments of the optimal triangulation. You should output the length by `cout << length << endl;` or `printf("%g\n",length)` where `length` is a `double` which stores the answer. If you're using Java, please output the first 7 characters of `new String(length)`.

Sample Input

```
2
4
0 0
0 1
1 1
1 0
4
0 0
1 1
2 1
1 0
```

Sample Output

```
1.41421
1
```