

I Identity Checker

You likely have seen that $x(\sin^2 x + \cos^2 x) - x = 0$, and you may have seen that $\sin(2x) - 2 \sin x \cos x = 0$. But did you know that $\tan(2x)(x - x \tan^2 x) - 2x \tan x = 0$? Would you believe that $\sin(2x) - 2 \cos x = 0$? That last one is false, but don't just take our word for it; you should write a program that determines whether an algebraic expression simplifies to zero (whenever it is defined).

I.1 Input

The input consists of multiple test cases, each on one line. Each test case starts with an integer N , the number of tokens that describes a formula. The next N tokens describe a formula in reverse polish notation. The notation works as follows. There is a stack that begins empty, and the following commands manipulate the contents of the stack:

- “**x**” pushes the variable x to the stack.
- “**sin**”, “**cos**”, and “**tan**” replace the top element of the stack with its sin, cos, and tan, respectively.
- “**+**”, “**-**”, and “*****” replace the top two elements of the stack (a on top, followed by b) with their sum ($b + a$), difference ($b - a$), and product ($b * a$), respectively.

You may assume that the input is valid, and results in a single item on the stack, which is the desired expression. The length of a line will be at most 300 characters. Function arguments can contain functions, so **x sin sin** is valid, but the recursion will not go any deeper than this. The input terminates with a line with $N = 0$. For example:

```
15 x sin x sin * x cos x cos * + x * x -
16 x sin x cos * x sin x cos * + x x + sin -
24 x x + tan x x tan x tan * x * - * x tan x * - x tan x * -
10 x x + sin x cos - x cos -
0
```

I.2 Output

For each test case, print out a single line that contains “**Identity**” if the expression is always zero, and “**Not an identity**” otherwise (quotes added for clarity). For example, the correct output for the sample input above would be:

```
Identity
Identity
Identity
Not an identity
```