5. Parse This!

Program Name: parse.java Input File: parse.in

Parse, parse, parse. You love to parse. You want to take a group of words and determine if they form a sentence. For the purposes of this problem, a sentence <S> can be defined in terms of its parts (noun phrases <NP>, verb phrases <VP>, prepositional phrases <PP>, and other parts of speech) using the following rules:

```
    <S> = <NP> <VP> [ <conjunction> <NP> <VP> ]
    <NP> = [ <article> ] [ <adjective> ] <noun>
    <VP> = <verb> [ <adverb> ] [ <PP> ]
    <PP> = <preposition> <NP>
    <noun> = I | bridge | boy | dog | pizza | home | ball | store
    <verb> = threw | ran | bought | eats | buy | loves | went
    <adjective> = big | tall | tasty | round | blue
    <adverb> = quickly | loudly
    <preposition> = to | under | from
    <article> = a | an | the
    <conjunction> = and | or | but | yet
```

Angle brackets indicate a placeholder that has its own rule defining it.
Square brackets indicate optional elements.
Vertical lines separate a list of choices, exactly one of which must be matched.
Note that the rules are case-insensitive ("the" is equivalent to "The").

Input

The first line of input will consist of a single integer, *n*, indicating the number of datasets in the remaining input. Each dataset will consist of:

1. A single line consisting solely of words and spaces (no punctuation), where a word is a contiguous unit of alphabetic characters.

Output

For each dataset in the input, output a single line with the phrase "Sentence" if the given line is a sentence according to the described rules. Otherwise, output a single line with the phrase "Not a sentence".

Example Input File

```
4
I went quickly to the big store and I bought
The dog eats quickly
George eats pizza
Tim eats and loves and threw
```

Example Output To Screen

```
Sentence
Sentence
Not a sentence
Not a sentence
```