

## C. Try and Catch

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Vasya is developing his own programming language VPL (Vasya Programming Language). Right now he is busy making the system of exceptions. He thinks that the system of exceptions must function like that.

The exceptions are processed by try-catch-blocks. There are two operators that work with the blocks:

1. The `try` operator. It opens a new try-catch-block.
2. The `catch(<exception_type>, <message>)` operator. It closes the try-catch-block that was started last and haven't yet been closed. This block can be activated only via exception of type `<exception_type>`. When we activate this block, the screen displays the `<message>`. If at the given moment there is no open try-catch-block, then we can't use the `catch` operator.

The exceptions can occur in the program in only one case: when we use the `throw` operator. The `throw(<exception_type>)` operator creates the exception of the given type.

Let's suggest that as a result of using some `throw` operator the program created an exception of type  $\alpha$ . In this case a try-catch-block is activated, such that this block's `try` operator was described in the program earlier than the used `throw` operator. Also, this block's `catch` operator was given an exception type  $\alpha$  as a parameter and this block's `catch` operator is described later than the used `throw` operator. If there are several such try-catch-blocks, then the system activates the block whose `catch` operator occurs earlier than others. If no try-catch-block was activated, then the screen displays message "Unhandled Exception".

To test the system, Vasya wrote a program that contains only `try`, `catch` and `throw` operators, one line contains no more than one operator, the whole program contains exactly one `throw` operator.

Your task is: given a program in VPL, determine, what message will be displayed on the screen.

### Input

The first line contains a single integer:  $n$  ( $1 \leq n \leq 10^5$ ) the number of lines in the program. Next  $n$  lines contain the program in language VPL. Each line contains no more than one operator. It means that input file can contain empty lines and lines, consisting only of spaces.

The program contains only operators `try`, `catch` and `throw`. It is guaranteed that the program is correct. It means that each started try-catch-block was closed, the `catch` operators aren't used unless there is an open try-catch-block. The program has exactly one `throw` operator. The program may have spaces at the beginning of a line, at the end of a line, before and after a bracket, a comma or a quote mark.

The exception type is a nonempty string, that consists only of upper and lower case english letters. The length of the string does not exceed 20 symbols. Message is a nonempty string, that consists only of upper and lower case english letters, digits and spaces. Message is surrounded with quote marks. Quote marks shouldn't be printed. The length of the string does not exceed 20 symbols.

Length of any line in the input file does not exceed 50 symbols.

### Output

Print the message the screen will show after the given program is executed.

### Examples

input
8 try

```
try
    throw ( AE )
catch ( BE, "BE in line 3")

try
    catch(AE, "AE in line 5")
catch(AE,"AE somewhere")
```

output

AE somewhere

input

```
8
try
    try
        throw ( AE )
    catch ( AE, "AE in line 3")

    try
        catch(BE, "BE in line 5")
    catch(AE,"AE somewhere")
```

output

AE in line 3

input

```
8
try
    try
        throw ( CE )
    catch ( BE, "BE in line 3")

    try
        catch(AE, "AE in line 5")
    catch(AE,"AE somewhere")
```

output

Unhandled Exception

Note

In the first sample there are 2 try-catch-blocks such that try operator is described earlier than throw operator and catch operator is described later than throw operator: try-catch(BE, "BE in line 3") and try-catch(AE, "AE somewhere") . Exception type is AE, so the second block will be activated, because operator catch(AE, "AE somewhere") has exception type AE as parameter and operator catch(BE, "BE in line 3") has exception type BE.

In the second sample there are 2 try-catch-blocks such that try operator is described earlier than throw operator and catch operator is described later than throw operator: try-catch(AE, "AE in line 3") and try-catch(AE, "AE somewhere") . Exception type is AE, so both blocks can be activated, but only the first one will be activated, because operator catch(AE, "AE in line 3") is described earlier than catch(AE, "AE somewhere")

In the third sample there is no blocks that can be activated by an exception of type CE.