Day 3: Throw

Objective

In this challenge, we practice using *throw* and *catch* statements to work with custom error messages.

Task

Complete the *isPositive* function below. It has one integer parameter, a. If the value of a is positive, it must return the string `YES`. Otherwise, it must *throw* an *Error* according to the following rules:

- If a is 0, *throw* an *Error* with $message =$ `Zero Error`.
- If a is negative, *throw* an *Error* with $message =$ `Negative Error`.

Input Format

Locked stub code in the editor reads the following input from stdin and passes each value of a to the function as an argument:
The first line is an integer, n, denoting the number of times the function will be called with some a.
Each line i of the n subsequent lines contains an integer denoting some a.

Constraints

- $1 \leq n \leq 5$
- $-100 \leq a \leq 100$

Output Format

If the value of a is positive, the function must return the string `YES`. Otherwise, it must *throw* an *Error* according to the following rules:

- If a is 0, *throw* an *Error* with $message =$ `Zero Error`.
- If a is negative, *throw* an *Error* with $message =$ `Negative Error`.

Sample Input 0

```
3
1
2
3
```

Sample Output 0

```
YES
YES
YES
```

Explanation 0

Each of the given values is positive, so we return `YES` each time. The value returned during each function call is printed on a new line by locked stub code in the editor.

Sample Input 1

```
3
2
0
6
```

Sample Output 1

```
YES
Zero Error
YES
```

Explanation 1

Locked stub code in the editor makes the following three calls to the *isPositive* function:

1. `isPositive(2)`: This returns `YES` because 2 is positive.
2. `isPositive(0)`: Because $a = 0$, we throw an *Error* with $message =$ `Zero Error`. This is caught by the locked stub code and the value of its $message$ is printed.
3. `isPositive(6)`: This returns `YES` because 6 is positive.

Sample Input 2

```
2
-1
20
```

Sample Output 2

```
Negative Error
YES
```

Explanation 2

Locked stub code in the editor makes the following two calls to the *isPositive* function:

1. `isPositive(-1)`: Because $a = -1$, we throw an *Error* with $message =$ `Negative Error`. This is caught by the locked stub code and the value of its $message$ is printed.
2. `isPositive(20)`: This returns `YES` because 20 is positive.