[CS 11 25.1] Lab 7i – Stage Puto

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

Problem Statement

There's puto bumbong on the stage! The students would like to get some of it.

There are n students in total, and they entered the stage one by one, forming one line from left to right. One can enter the stage in two ways: from the left, or from the right. Each student went to the stage from one of these sides and then inserted themselves to the forming line from the direction they entered.

Please maintain the list of students as they enter the stage to get some puto bumbong.

Task Details

Your task is to implement a *class* called `Stage`. An instance of this class must maintain a row of students as they enter a stage.

Its constructor takes in zero arguments and must initialize an empty row.

The class must implement the following methods:

- `.insert_right(s)` where `s` is a `str` denoting a student's name. This inserts student `s` to the right of the row. The method must not return anything.
- `.insert_left(s)` where `s` is a `str` denoting a student's name. This inserts student `s` to the left of the row. The method must not return anything.
- `.get(i)` where `i` is an `int`. if `0 <= i < n` where `n` is the current length of the row, then the method must return the name of student at index `i` from the left, using zero-indexing; otherwise, this method must raise a `ValueError`.

Restrictions

Note that some names are banned. Here are a few of them: `input`, `type`. This is not an exhaustive list. (If you accidentally use a variable name that turns out to be banned, please rename it.)

The following names are allowed: `map`, `filter`.

The following imports are allowed:

- `count`, `islice`, `chain`, `takewhile`, `starmap` and `zip_longest` from `itertools`.
- `cache`, `lru_cache`, `total_ordering`, `partial`, `reduce` and `wraps` from `functools`.
- `randint`, `randrange` and `choice` from `random`.
- `Fraction` from `fractions`.
- `dataclass` from `dataclasses`.
- `contextmanager` from `contextlib`.
- `Enum`, `auto` from `enum`.

(Read the docs to learn what they do!)

Anonymous functions are allowed.

Inner functions are allowed.

Classes, dataclasses and enums are allowed.

For this problem in particular:

- The source code limit is 3000.

Example Testing

Note: This assumes that your submission has filename `hope3c1.py`. Write this testing code in a separate file, say `test_hope3c1.py`, and run it to test your code.

Example 1 Testing

Copy

```
from contextlib import contextmanager

from hope3c1 import Stage

@contextmanager
def raises_value_error():
    """ context block must raise a ValueError """
    try:
        yield
    except ValueError:
        pass
    else:
        assert False


data = Stage()

data.insert_left('ren')
assert data.get(0) == 'ren'

data.insert_left('gino')
data.insert_right('jem')
assert data.get(0) == 'gino'
assert data.get(1) == 'ren'
assert data.get(2) == 'jem'
with raises_value_error():
    data.get(3)

data.insert_left('carlo')
data.insert_right('kevin')
data.insert_left('kelvin')
assert data.get(1) == 'carlo'
assert data.get(0) == 'kelvin'
assert data.get(5) == 'kevin'
with raises_value_error():
    data.get(-1)
```

Constraints

Here, n is the total number of method calls.

- The function `Stage` will be called at most $60{,}000$ times.
- The sum of ns across all calls will be $\leq 200{,}000$.
- Each name is a nonempty string of at most 6 lowercase English letters.
- $|i| \leq 10^6$

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 70 ❤️ points if you solve all test cases where:
  - The sum of the ns across all calls will be 50.
  - `.insert_left` will not be called.
- You get 85 🔴 points if you solve all test cases where:
  - `.insert_left` will not be called.
- You get 85 🔴 points if you solve all test cases where:
  - The sum of the ns across all calls will be 50.
- You get 50 🔴 points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.