



# [CS 11 25.1] Lab 3d – Stage Photo

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

Submit solution

[CS 11 25.1]

Lab Exercise 3

My submissions

✓ Points: 175 (partial)

⌚ Time limit: 4.0s

≡ Memory limit: 1G

➤ Problem type

▼ Allowed languages

py3

The graduation ceremony has just finished! A bunch of students have decided to go to the stage to take an unofficial photo for fun.

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.

Given the names of the  $n$  students who entered the stage in chronological order, as well as which side they entered the stage from, what is the sequence of students from left to right?

## Task Details

Your task is to implement a function called `stage_sequence`. The function takes a single argument, a `tuple` of  $n$  pairs representing the students entering the stage in chronological order. Each pair consists of two `str`s:

- the first `str` denotes the name of the student.
- the second `str` is either `"left"` or `"right"` depending on which side they entered the stage.

The function must return a `tuple` of `str`s denoting the sequence of student names from left to right.

## Restrictions

Note that many names are banned. Here are a few of them: `zip`, `input`, `type`, `reversed`. 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 now allowed: `range`, `list`, `enumerate`, `print`, `append`, `pop`, `extend`, `remove`, `sort`, `insert`, `clear`, `reverse`, `reversed`.

For this problem:

- Loops and lists are allowed.
- Up to 4 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The source code limit is 600.

## Example Calls

### Example 1 Function Call

```
stage_sequence((  
    ('ren', 'left'),  
    ('juancho', 'left'),  
    ('jem', 'right'),  
    ('carlo', 'left'),  
    ('kevin', 'right'),  
    ('kelvin', 'left'),  
)
```

Copy

### Example 1 Return Value

```
('kelvin', 'carlo', 'juancho', 'ren', 'jem', 'kevin')
```

Copy

## Constraints

- The function `stage_sequence` will be called at most 50,000 times.
- The sum of  $n$  across all inputs will be at most 150,000.
- $0 \leq n \leq 150,000$
- Each name is a nonempty string of between 1 and 6 lowercase letters.

## 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 100 ● points if you solve all test cases where:
  - $n \leq 50$
  - the sum of all  $n$  is  $\leq 500$ .
- You get 50 ● points if you solve all test cases where:
  - $n \leq 4,000$
  - the sum of all  $n$  is  $\leq 8,000$ .
- You get 25 ● points if you solve all test cases.

## Clarifications

Report an issue

No clarifications have been made at this time.