[CS 11] Prac 3b – Shaman

oj.dcs.upd.edu.ph/problem/cs11prac3b

Problem Statement

Endugu is a shaman who resides in a dungeon hidden deep in the jungle. Unbeknownst to most, he has a day job of being a jeepney operator.

Today is a good day—there are so many passengers! They are in line waiting for a jeepney.

Endugu's task is to split the passenger queue into several groups, where each group will ride together in a jeepney.

Some jeepneys can accomodate up to 99 passengers on both sides (that's why Endugu often shouts "shaman yan!"), so they can accommodate up to 2020 passengers overall (if we count the seats in front). However, some jeepneys may be able to accommodate less.

For simplicity, let's assume that every jeepney can accommodate exactly pp passengers.

Given the sequence of passengers, please split them up into their corresponding jeepney groups, i.e., groups of size pp. It is okay for the last group to have less than pp passengers, if there aren't enough to reach pp. However, please keep the order of the passengers the same.

Task Details

Your task is to implement a function called `jeepney_groups`. This function has two parameters:

- `passengers` — a `tuple` of `str`s denoting the sequence of passenger names in their order in the queue.
- `p` — an `int` denoting the number of passengers each jeepney can accommodate.

In particular, your function will be declared as follows:

Copy

```
def jeepney_groups(passengers, p):
```

The function must return a `tuple` of `tuple`s. Each `tuple` must consists of pp `str`s, except possibly the last, and denotes a group of passengers.

Restrictions

For this problem:

- Recursion is **disallowed**.
- Additional functions are **disallowed**.
- Comprehensions are allowed.
- The `range`, `min`, `max`, and `sum` symbols are allowed.
- The source code limit is 250250.

Example Calls

Example 1 Function Call

Copy

```
jeepney_groups(('eiko', 'dagger', 'freya', 'amarant', 'vivi',
'steiner', 'zidane', 'quina'), 2)
```

Example 1 Return Value

Copy

```
(
    ('eiko', 'dagger'),
    ('freya', 'amarant'),
    ('vivi', 'steiner'),
    ('zidane', 'quina'),
)
```

Example 2 Function Call

Copy

```
jeepney_groups(('eiko', 'dagger', 'freya', 'amarant', 'vivi',
'steiner', 'zidane', 'quina'), 3)
```

Example 2 Return Value

Copy

```
(
    ('eiko', 'dagger', 'freya'),
    ('amarant', 'vivi', 'steiner'),
    ('zidane', 'quina'),
)
```

Constraints

- The function `jeepney_groups` will be called at most 200200 times.
- `passengers` will have at most 5050 elements.
- $1 \leq p \leq 201 \leq p \leq 20$
- Each name is nonempty and consists of up to 1010 lowercase English letters.

Scoring

- You get 100100 ❤️ points if you solve all test cases where:
  - pp divides the number of passengers.
- You get 5050 ❤️ points if you solve all test cases.

Report an issue

Clarifications

No clarifications have been made at this time.