



[CS 11 25.1] Lab 2j – Finals

Problem Statement

[Submit solution](#)

In one of your CS courses, there is a *pre-final* policy that goes something along the lines of:

If your pre-final grade is ≥ 60 , you do not need to take the finals.

Of course, students that have dropped the course also don't need to take the finals.

This seems rather straightforward to automate! Can you try it out?

[My submissions](#)**Points:** 275 (partial)**Time limit:** 3.0s**Memory limit:** 1G

Task Details

Your task is to implement a function named `get_finals_takers`, which should have the following *signature*:

```
def get_finals_takers(prefinal_grades, droppers):
```

[Copy](#)

The above says that it has two arguments `prefinal_grades` and `droppers`.

- `prefinal_grades` is a `tuple` of pairs (`tuple`s of length 2), where each pair contains a name of a student (as a `str`) and their prefinal grade (as an `int`).
- `droppers` is a `tuple` of strings (`str`s) denoting the students that have dropped the course.

The function must return a `frozenset` of strings denoting the students that need to take the final exam.

Restrictions

- The following symbols can now be used: `min`, `max`, `sum`, `range`, `all`, `any`.
- recursion is *disallowed*.
- comprehensions are allowed.
- at most 6 functions can be defined.
- Your source code must have at most 800 bytes.

Examples

Example 1 Function Call

```
get_finals_takers((("Daryll", 50), ("Jem", 75), ("Kevin", 100), ("Kevin",)))
```

[Copy](#)

Example 1 Return Value

```
frozenset(("Daryll",))
```

[Copy](#)

Constraints

- The function `get_finals_takers` will be called at most 100 times.
- Each prefinal grade is between 0 and 100, inclusive.
- Each name is a nonempty string of (uppercase and lowercase) English letters with length at most 8.
- The sum of the lengths of `prefinal_grades` across all calls to `get_finals_takers` will be at most 200,000.
- Each name in `droppers` appears as a name in `prefinal_grades`.
- The names in `prefinal_grades` are unique.
- The names in `droppers` are unique.

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 50 ❤ points if you solve all test cases where:
 - The sum of the lengths of `prefinal_grades` across all calls to `get_finals_takers` will be at most 100.
 - All prefinal grades are less than 60.

- You get 200 🎯 points if you solve all test cases where:
 - The sum of the lengths of `prefinal_grades` across all calls to `get_finals_takers` will be at most 100.

- You get 25 💕 points if you solve all test cases.

Clarifications

[Report an issue](#)

No clarifications have been made at this time.