[CS 11] Prac 1i – Pares ng Pares

oj.dcs.upd.edu.ph/problem/cs11prac1i

Problem Statement

There are nn consecutive food stalls, each serving their own version of *beef pares*. You want to buy two beef pares servings that you will share with your spouse.

For scientific purposes, you want to buy two pares servings from two different stalls.

It can easily be shown that there are n(n−1)/2$n\ (\ n-1\ )\ /\ 2$ ways to do so, ignoring order.

Each stall serves beef pares at their own price.

For each of the n(n−1)/2$n\ (\ n-1\ )\ /\ 2$ ways to buy a pair of beef pares, what do you need to pay?

Please return all of these answers as a `tuple` of pairs, each pair denoting the prices of the pair of beef pares servings, in any order. The pairs themselves may also be enumerated in any order. Note that in particular, the return value will consist of n(n−1)/2$n\ (\ n-1\ )\ /\ 2$ pairs.

Task Details

Your task is to implement a function called `pares_pairs`. This function has a single parameter, a `tuple` of nn `int`s denoting the prices that the nn stalls are selling their beef pares at.

The function must return a `tuple` of n(n−1)/2$n\ (\ n-1\ )\ /\ 2$ pairs of `int`s, as described in the problem statement.

Restrictions

For this problem:

- Assignment is allowed.
- Recursion is allowed.
- Up to 66 function definitions are allowed.
- Comprehensions are **disallowed**.
- `range` is **disallowed**.
- The `abs` symbol is now allowed.

- The source code limit is 10001000.

Example Calls

Example 1 Function Call

Copy

```
pares_pairs((3, 1, 4))
```

Example 1 Return Value

Copy

```
((3, 1), (1, 4), (3, 4))
```

Note that other orderings of these 33 pairs are allowed. The pairs themselves may also have their elements swapped.

Example 2 Function Call

Copy

```
pares_pairs(())
```

Example 2 Return Value

Copy

```
()
```

Constraints

- The function `pares_pairs` will be called at most 1,0001,000 times.
- The argument contains at most 100100 elements, each of which is a positive integer at most 100100.

Scoring

You get 100100 ❤️ points if you solve all test cases.
Report an issue

Clarifications

No clarifications have been made at this time.