



# [CS 11] Prac 7h – k-Subslices

## Problem Statement

Given an integer  $k$  and a sequence of integers, give all of its  $k$ -tuples of elements in consecutive locations.

## Task Details

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

- the first is the `int`  $k$ .
- the second is an iterable of `int`s.

The function must return a *generator* that generates  $k$ -tuples of `int`s, as described in the problem statement.

Note that your generator must be **as lazy as possible**. It should yield each resulting next element as soon as it has enough information, and it should produce these results while advancing the input generators for as little as possible.

## Restrictions

(See 7a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 8 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 700.

## Example Calls

### Example 1 Function Call

```
[*segments(4, (3, 1, 4, 1, 5, 9))]
```

Copy

### Example 1 Return Value

```
[(3, 1, 4, 1), (1, 4, 1, 5), (4, 1, 5, 9)]
```

Copy

### Example 2 Function Call

```
[*segments(40, (3, 1, 4, 1, 5, 9))]
```

Copy

### Example 2 Return Value

```
[]
```

Copy

## Constraints

When your program is run:

- The function `segments` will be called at most 50 times.
- At most 100 elements will be consumed from the returned generator.
- Each element of the input sequence is a positive integer at most  $10^5$ .
- $1 \leq k \leq 10^6$

## Scoring

- You get 120 ❤ points if you solve all test cases.

## Clarifications

Report an issue

No clarifications have been made at this time.