

# [CS 11] Prac 7b – Add To All

## Problem Statement

Given a sequence of integers and an integer  $a$ , give the same sequence but with every integer increased by  $a$ .

## Task Details

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

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

The function must return a *generator* that generates `int`s. The `int`s must be the elements of the second argument but each increased by  $a$ .

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 300.

## Example Calls

### Example 1 Function Call

```
[*increased_by(10, (3, 1, 4))]
```

### Example 1 Return Value

```
[13, 11, 14]
```

### Example 2 Function Call

```
print(*increased_by(10, iter((3, 1, 4))))
```

### Example 2 Output


```
13 11 14
```

## Constraints

When your program is run:

- The function `increased_by` will be called at most 200 times.
- At most 500 elements will be consumed from the returned generator.
- Each element of the input sequence is a positive integer at most  $10^{10}$ .
- $1 \leq a \leq 10^{10}$ .

## Scoring

- You get 100  points if you solve all test cases.


## Clarifications


No clarifications have been made at this time.

Report an issue

Submit solution

[CS 11]

Practice 7 

[My submissions](#) 

✓ **Points:** 100 (partial)

⌚ **Time limit:** 6.0s

📄 **Memory limit:** 1G

✎ **Author:**  
kvatienza (Kevin Atienza)

➤ **Problem type**

✓ **Allowed languages**  
NONE, py3