Question1

Create a function that takes three parameters where:

- `x` is the start of the range (inclusive).
- `y` is the end of the range (inclusive).
- `n` is the divisor to be checked against.

Return an ordered list with numbers in the range that are divisible by the third parameter `n`. Return an empty list if there are no numbers that are divisible by `n`.

Examples

```
list_operation(1, 10, 3) ➞ [3, 6, 9]

list_operation(7, 9, 2) ➞ [8]

list_operation(15, 20, 7) ➞ []
```

Question2

Create a function that takes in two lists and returns `True` if the second list follows the first list by **one** element, and `False` otherwise. In other words, determine if the second list is the first list shifted to the right by 1.

Examples

```
simon_says([1, 2], [5, 1]) ➞ True

simon_says([1, 2], [5, 5]) ➞ False

simon_says([1, 2, 3, 4, 5], [0, 1, 2, 3, 4]) ➞ True

simon_says([1, 2, 3, 4, 5], [5, 5, 1, 2, 3]) ➞ False
```

Notes

- Both input lists will be of the same length, and will have a minimum length of 2.
- The values of the `0`-indexed element in the second list and the `n-1`th indexed element in the first list do not matter.

Question3

A group of friends have decided to start a secret society. The name will be the first letter of each of their names, sorted in alphabetical order.

Create a function that takes in a list of names and returns the name of the secret society.

Examples

```
society_name(["Adam", "Sarah", "Malcolm"]) ➞ "AMS"

society_name(["Harry", "Newt", "Luna", "Cho"]) ➞ "CHLN"

society_name(["Phoebe", "Chandler", "Rachel", "Ross", "Monica", "Joey"])
```

Question4

An isogram is a word that has no duplicate letters. Create a function that takes a string and returns either `True` or `False` depending on whether or not it's an "isogram".

Examples

```
is_isogram("Algorism") ➞ True

is_isogram("PasSword") ➞ False
# Not case sensitive.

is_isogram("Consecutive") ➞ False
```

Notes

- Ignore letter case (should not be case sensitive).
- All test cases contain valid one word strings.

Question5

Create a function that takes a string and returns `True` or `False`, depending on whether the characters are in order or not.

Examples

```
is_in_order("abc") ➞ True

is_in_order("edabit") ➞ False

is_in_order("123") ➞ True

is_in_order("xyzz") ➞ True
```

Notes

You don't have to handle empty strings.