Question 1

Create a function that takes a list of non-negative **integers** and **strings** and return a new list without the strings.

**Examples**

filter\_list([1, 2, "a", "b"]) ➞ [1, 2]

filter\_list([1, "a", "b", 0, 15]) ➞ [1, 0, 15]

filter\_list([1, 2, "aasf", "1", "123", 123]) ➞ [1, 2, 123]

def filter\_list(lst):

return [x for x in lst if isinstance(x, int)]

# Example usage

print(filter\_list([1, 2, "a", "b"])) # Output: [1, 2]

print(filter\_list([1, "a", "b", 0, 15])) # Output: [1, 0, 15]

print(filter\_list([1, 2, "aasf", "1", "123", 123])) # Output: [1, 2, 123]

Question 2

The "Reverser" takes a string as input and returns that string in reverse order, with the opposite case.

### Examples

reverse("Hello World") ➞ "DLROw OLLEh"

reverse("ReVeRsE") ➞ "eSrEvEr"

reverse("Radar") ➞ "RADAr"

def reverse(string):

reversed\_string = string[::-1]

return reversed\_string.swapcase()

# Example usage

print(reverse("Hello World")) # Output: "DLROw OLLEh"

print(reverse("ReVeRsE")) # Output: "eSrEvEr"

print(reverse("Radar")) # Output: "RADAr"

Question 3

You can assign variables from lists like this:

lst = [1, 2, 3, 4, 5, 6]

first = lst[0]

middle = lst[1:-1]

last = lst[-1]

print(first) ➞ outputs 1

print(middle) ➞ outputs [2, 3, 4, 5]

print(last) ➞ outputs 6

With Python 3, you can assign variables from lists in a much more succinct way. Create variables first, middle and last from the given list using **destructuring assignment** (check the **Resources** tab for some examples), where:

first ➞ 1

middle ➞ [2, 3, 4, 5]

last ➞ 6

Your task is to unpack the list writeyourcodehere into three variables, being first, middle, and last, with middle being everything in between the first and last element. Then print all three variables.

lst = [1, 2, 3, 4, 5, 6]

first, \*middle, last = lst

print(first) # Output: 1

print(middle) # Output: [2, 3, 4, 5]

print(last) # Output: 6

Question 4

Write a function that calculates the **factorial** of a number **recursively**.

### Examples

factorial(5) ➞ 120

factorial(3) ➞ 6

factorial(1) ➞ 1

factorial(0) ➞ 1

def factorial(n):

# Base case: factorial of 0 and 1 is 1

if n == 0 or n == 1:

return 1

# Recursive case: factorial of n is n multiplied by factorial of (n-1)

return n \* factorial(n - 1)

print(factorial(5)) # Output: 120

print(factorial(3)) # Output: 6

print(factorial(1)) # Output: 1

print(factorial(0)) # Output: 1

Question 5

Write a function that moves all elements of one type to the **end** of the list.

### Examples

move\_to\_end([1, 3, 2, 4, 4, 1], 1) ➞ [3, 2, 4, 4, 1, 1]

# Move all the 1s to the end of the array.

move\_to\_end([7, 8, 9, 1, 2, 3, 4], 9) ➞ [7, 8, 1, 2, 3, 4, 9]

move\_to\_end(["a", "a", "a", "b"], "a") ➞ ["b", "a", "a", "a"]

def move\_to\_end(lst, element):

# Create two empty lists for elements of other types and elements of the specified type

other\_elements = []

specified\_elements = []

# Iterate through the list and separate the elements

for item in lst:

if item == element:

specified\_elements.append(item)

else:

other\_elements.append(item)

# Combine the two lists with the specified elements at the end

return other\_elements + specified\_elements

print(move\_to\_end([1, 3, 2, 4, 4, 1], 1)) # Output: [3, 2, 4, 4, 1, 1]

print(move\_to\_end([7, 8, 9, 1, 2, 3, 4], 9)) # Output: [7, 8, 1, 2, 3, 4, 9]

print(move\_to\_end(["a", "a", "a", "b"], "a")) # Output: ["b", "a", "a", "a"]