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]

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def filter\_list(my\_list):

my\_ints = []

for i in my\_list:

if type(i) == int:

my\_ints.append(i)

print(my\_ints)

return my\_ints

try:

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

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

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

lg.info("""Class factorial() has been called has been called""")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

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"

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def reverse(my\_string):

my\_reversed = ""

for i in my\_string[::-1]:

my\_reversed += i

print(my\_reversed.swapcase())

try:

reverse("Hello World")

reverse("ReVeRsE")

reverse("Radar")

lg.info("""Class factorial() has been called has been called""")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

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.

Ans.

my\_list = [1,2,3,4,5,6]

first, \*middle, last = my\_list

print(first)

print(middle)

print(last)

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

Ans.

def factor(n):

if n == 0:

return 0

if n == 1:

return 1

else:

return n\*factor(n-1)

factor(5)

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"]

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def move\_to\_end(my\_list, choice):

my\_list\_copy = my\_list.copy()

if choice not in my\_list:

print("your given choice does not exist in main list")

else:

for i in range(my\_list.count(choice)):

my\_list\_copy.remove(choice)

for j in range(my\_list.count(choice)):

my\_list\_copy.append(choice)

print(my\_list\_copy)

try:

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

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

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

lg.info("""Class factorial() has been called has been called""")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass