Question1

Write a function that takes a list and a number as arguments. Add the number to the end of the list, then remove the first element of the list. The function should then return the updated list.

**Examples**

next\_in\_line([5, 6, 7, 8, 9], 1) ➞ [6, 7, 8, 9, 1]

next\_in\_line([7, 6, 3, 23, 17], 10) ➞ [6, 3, 23, 17, 10]

next\_in\_line([1, 10, 20, 42 ], 6) ➞ [10, 20, 42, 6]

next\_in\_line([], 6) ➞ "No list has been selected"

# Write a function that takes a list and a number as arguments.

# Add the number to the end of the list, then remove the first element of the list.

# The function should then return the updated list. Examples :

# next\_in\_line([5, 6, 7, 8, 9], 1) ➞ [6, 7, 8, 9, 1]

# next\_in\_line([7, 6, 3, 23, 17], 10) ➞ [6, 3, 23, 17, 10]

# next\_in\_line([1, 10, 20, 42 ], 6) ➞ [10, 20, 42, 6]

# next\_in\_line([], 6) ➞ "No list has been selected"

# Define a function which takes list and number as arguments and returns a list

def function1(l1, num):

if len(l1) == 0:

return "No list has been selected"

else:

l1.append(num)

l1.remove(l1[0])

return l1

# Check with different parameters

l1 = function1([5, 6, 7, 8, 9], 1)

print(l1)

l1 = function1([7, 6, 3, 23, 17], 10)

print(l1)

l1 = function1([1, 10, 20, 42 ], 6)

print(l1)

l1 = function1([], 6)

print(l1)

Question2

Create the function that takes a list of dictionaries and returns the sum of people's budgets.

### Examples

get\_budgets([

{ "name": "John", "age": 21, "budget": 23000 },

{ "name": "Steve", "age": 32, "budget": 40000 },

{ "name": "Martin", "age": 16, "budget": 2700 }

]) ➞ 65700

get\_budgets([

{ "name": "John", "age": 21, "budget": 29000 },

{ "name": "Steve", "age": 32, "budget": 32000 },

{ "name": "Martin", "age": 16, "budget": 1600 }

]) ➞ 62600

# Create the function that takes a list of dictionaries and returns the sum of people's budgets.

# Create two sample lists for testing purpose

list1 = [

{"name": "John", "age": 21, "budget": 23000},

{"name": "Steve", "age": 32, "budget": 40000},

{"name": "Martin", "age": 16, "budget": 2700}

]

list2 = [

{"name": "John", "age": 21, "budget": 29000},

{"name": "Steve", "age": 32, "budget": 32000},

{"name": "Martin", "age": 16, "budget": 1600}

]

# Define a function to get the sum of budgets in dictionaries in a list object

def function1(l1):

sum = 0

for i in l1:

sum = sum + int(i.get("budget"))

return sum

# Call the function, function1 with list inputs list1, list2 and check the result

print("Sum of people's budget is ", function1(list1))

print("Sum of people's budget is ", function1(list2))

Question3

Create a function that takes a string and returns a string with its letters in alphabetical order.

### Examples

alphabet\_soup("hello") ➞ "ehllo"

alphabet\_soup("edabit") ➞ "abdeit"

alphabet\_soup("hacker") ➞ "acehkr"

alphabet\_soup("geek") ➞ "eegk"

alphabet\_soup("javascript") ➞ "aacijprstv"

# Create a function that takes a string and returns a string with its letters in alphabetical order.

# Examples alphabet\_soup("hello") ➞ "ehllo"

# alphabet\_soup("edabit") ➞ "abdeit"

# alphabet\_soup("hacker") ➞ "acehkr"

# alphabet\_soup("geek") ➞ "eegk"

# alphabet\_soup("javascript") ➞ "aacijprstv"

# Declare a string object

string1 = "python"

# Define a function function1 to sort string in alphabetical order

# Use sorted inbuilt string function to sort string object

# and join the characters using join function

def function1(str1):

return ''.join(sorted(str1))

# pass string object into the function, function1() and print the output

print(function1(string1))

Question4

Suppose that you invest $10,000 for 10 years at an interest rate of 6% compounded monthly. What will be the value of your investment at the end of the 10 year period?

Create a function that accepts the principal p, the term in years t, the interest rate r, and the number of compounding periods per year n. The function returns the value at the end of term rounded to the nearest cent.

For the example above:

compound\_interest(10000, 10, 0.06, 12) ➞ 18193.97

Note that the interest rate is given as a decimal and n=12 because with monthly compounding there are 12 periods per year. Compounding can also be done annually, quarterly, weekly, or daily.

### Examples

compound\_interest(100, 1, 0.05, 1) ➞ 105.0

compound\_interest(3500, 15, 0.1, 4) ➞ 15399.26

compound\_interest(100000, 20, 0.15, 365) ➞ 2007316.26

# Create a function that accepts the principal p, the term in years t, the interest rate r,

# and the number of compounding periods per year n. The function returns the value at the end of term rounded to the nearest cent.

# compound\_interest(10000, 10, 0.06, 12) ➞ 18193.97

# Examples compound\_interest(100, 1, 0.05, 1) ➞ 105.0

# compound\_interest(3500, 15, 0.01, 4) ➞ 15399.26

# compound\_interest(100000, 20, 0.15, 365) ➞ 2007316.26

# Declare a function to calculate compound interest

def function1(p,t,r,n):

return round(p\*(pow((1+r/n),n\*t)),2)

# Call the function with different inputs provided and check the output

print(function1(10000, 10, 0.06, 12)) # output should be 18193.97

print(function1(100, 1, 0.05, 1)) # output should be 105.0

print(function1(3500, 15, 0.1, 4)) # output should be 15399.26

print(function1(100000, 20, 0.15, 365)) # ouput should be 2007316.26

Question5

Write a function that takes a list of elements and returns only the integers.

### Examples

return\_only\_integer([9, 2, "space", "car", "lion", 16]) ➞ [9, 2, 16]

return\_only\_integer(["hello", 81, "basketball", 123, "fox"]) ➞ [81, 123]

return\_only\_integer([10, "121", 56, 20, "car", 3, "lion"]) ➞ [10, 56, 20, 3]

return\_only\_integer(["String", True, 3.3, 1]) ➞ [1]

# Write a function that takes a list of elements and returns only the integers.

# Examples return\_only\_integer([9, 2, "space", "car", "lion", 16]) ➞ [9, 2, 16]

# return\_only\_integer(["hello", 81, "basketball", 123, "fox"]) ➞ [81, 123]

# return\_only\_integer([10, "121", 56, 20, "car", 3, "lion"]) ➞ [10, 56, 20, 3]

# return\_only\_integer(["String", True, 3.3, 1]) ➞ [1]

# Define a function, function1 to get only integers from a list of elements

def function1(l1):

l2 = []

for i in l1:

if type(i) == int:

l2.append(i)

return l2

# Check with multiple inputs and check the results

print(function1([10, "121", 56, 20, "car", 3, "lion"]))

print(function1(["hello", 81, "basketball", 123, "fox"]))

print(function1([10, "121", 56, 20, "car", 3, "lion"]))

print(function1(["String", True, 3.3, 1]))