1. def next\_in\_line(in\_list,in\_num):

if len(in\_list) > 1:

in\_list.append(in\_num)

in\_list.remove(in\_list[0])

print(f'Output ➞ {in\_list}')

else:

print('No list has been selected')

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

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

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

next\_in\_line([], 6)

Output:

Output ➞ [6, 7, 8, 9, 1]

Output ➞ [6, 3, 23, 17, 10]

Output ➞ [10, 20, 42, 6]

No list has been selected

1. def get\_budgets(in\_dict):

sum = 0

for ele in in\_dict:

sum += ele["budget"]

print(f'Output ➞ {sum}')

get\_budgets([

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

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

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

])

get\_budgets([

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

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

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

])

Output:

Output ➞ 65700

Output ➞ 62600

1. def alphabet\_soup(in\_string):

out\_string = ''.join(sorted(in\_string))

print(f'{in\_string} ➞ {out\_string}')

alphabet\_soup("hello")

alphabet\_soup("edabit")

alphabet\_soup("hacker")

alphabet\_soup("geek")

alphabet\_soup("javascript")

Output:

hello ➞ ehllo

edabit ➞ abdeit

hacker ➞ acehkr

geek ➞ eegk

javascript ➞ aacijprstv

1. def compound\_interest(principal,years,roi,cp):

ci = principal\*(1+(roi/cp))\*\*(cp\*years)

print(f'Output ➞ {ci:.2f}')

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

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

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

Output:

Output ➞ 105.00

Output ➞ 15399.26

Output ➞ 2007316.26

1. def return\_only\_integer(in\_list):

out\_list = []

for ele in in\_list:

if type(ele) == int:

out\_list.append(ele)

print(f'{in\_list} ➞ {out\_list}')

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

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

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

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

Output:

[9, 2, 'space', 'car', 'lion', 16] ➞ [9, 2, 16]

['hello', 81, 'basketball', 123, 'fox'] ➞ [81, 123]

[10, '121', 56, 20, 'car', 3, 'lion'] ➞ [10, 56, 20, 3]

['String', True, 3.3, 1] ➞ [1]