1. def track\_robot(in\_list):

output = [0,0]

for ele in in\_list:

ele = ele.split(" ")

if ele[0] in ['right','left']:

output[0] = output[0]-int(ele[1]) if ele[0] == 'left' else output[0]+int(ele[1])

else:

output[1] = output[1]-int(ele[1]) if ele[0] == 'down' else output[1]+int(ele[1])

print(f'track\_robot({in\_list}) ➞ {output}')

track\_robot(["right 10", "up 50", "left 30", "down 10"])

track\_robot([])

track\_robot(["right 100", "right 100", "up 500", "up 10000"])

Output:

track\_robot(['right 10', 'up 50', 'left 30', 'down 10']) ➞ [-20, 40]

track\_robot([]) ➞ [0, 0]

track\_robot(['right 100', 'right 100', 'up 500', 'up 10000']) ➞ [200, 10500]

1. def find\_longest(in\_string):

len\_list = []

for ele in in\_string.split(' '):

len\_list.append(len(ele))

print(f'find\_longest({in\_string}) ➞ {in\_string.split(" ")[len\_list.index(max(len\_list))].lower()}')

find\_longest("A thing of beauty is a joy forever.")

find\_longest("Forgetfulness is by all means powerless!")

find\_longest("\"Strengths\" is the longest and most commonly used word that contains only a single vowel.")

Output:

find\_longest(A thing of beauty is a joy forever.) ➞ forever.

find\_longest(Forgetfulness is by all means powerless!) ➞ forgetfulness

find\_longest("Strengths" is the longest and most commonly used word that contains only a single vowel.) ➞ "strengths"

1. def interview(in\_list, in\_time):

output = 'qualified'

if in\_time > 120 or max(in\_list[0:2]) > 5 or max(in\_list[2:4]) > 10 or max(in\_list[4:6]) > 15 or max(in\_list[6:8]) > 20 or len(in\_list) != 8:

output = 'disqualified'

print(f'interview{in\_list,in\_time} ➞ {output}')

interview([5, 5, 10, 10, 15, 15, 20, 20], 120)

interview([2, 3, 8, 6, 5, 12, 10, 18], 64)

interview([5, 5, 10, 10, 25, 15, 20, 20], 120)

interview([5, 5, 10, 10, 15, 15, 20], 120)

interview([5, 5, 10, 10, 15, 15, 20, 20], 130)

Output:

interview([5, 5, 10, 10, 15, 15, 20, 20], 120) ➞ qualified

interview([2, 3, 8, 6, 5, 12, 10, 18], 64) ➞ qualified

interview([5, 5, 10, 10, 25, 15, 20, 20], 120) ➞ disqualified

interview([5, 5, 10, 10, 15, 15, 20], 120) ➞ disqualified

interview([5, 5, 10, 10, 15, 15, 20, 20], 130) ➞ disqualified

1. def chunkify(in\_list,chunk\_size):

output = []

for i in range(0,len(in\_list),chunk\_size):

output.append(in\_list[i:i+chunk\_size])

print(f'chunkify{in\_list, chunk\_size} ➞ {output}')

chunkify([2, 3, 4, 5], 2)

chunkify([2, 3, 4, 5, 6], 2)

chunkify([2, 3, 4, 5, 6, 7], 3)

chunkify([2, 3, 4, 5, 6, 7], 1)

chunkify([2, 3, 4, 5, 6, 7], 7)

Output:

chunkify([2, 3, 4, 5], 2) ➞ [[2, 3], [4, 5]]

chunkify([2, 3, 4, 5, 6], 2) ➞ [[2, 3], [4, 5], [6]]

chunkify([2, 3, 4, 5, 6, 7], 3) ➞ [[2, 3, 4], [5, 6, 7]]

chunkify([2, 3, 4, 5, 6, 7], 1) ➞ [[2], [3], [4], [5], [6], [7]]

chunkify([2, 3, 4, 5, 6, 7], 7) ➞ [[2, 3, 4, 5, 6, 7]]

1. def get\_prices(in\_list):

out\_list = []

for ele in in\_list:

out\_list.append(float((ele.split('$')[-1]).split(')')[0]))

print(f'get\_prices({in\_list}) ➞ {out\_list}')

get\_prices(["salad ($4.99)"])

get\_prices(["artichokes ($1.99)","rotiserrie chicken ($5.99)","gum ($0.75)"])

get\_prices(["ice cream ($5.99)","banana ($0.20)","sandwich ($8.50)","soup ($1.99)"])

Output:

get\_prices(['salad ($4.99)']) ➞ [4.99]

get\_prices(['artichokes ($1.99)', 'rotiserrie chicken ($5.99)', 'gum ($0.75)']) ➞ [1.99, 5.99, 0.75]

get\_prices(['ice cream ($5.99)', 'banana ($0.20)', 'sandwich ($8.50)', 'soup ($1.99)']) ➞ [5.99, 0.2, 8.5, 1.99]