Assignment3

# Q1.1: Write a Python Program to implement your own myreduce() function which works exactly like Python's built-in function reduce()

# Answer :

def myreduce(a, b):

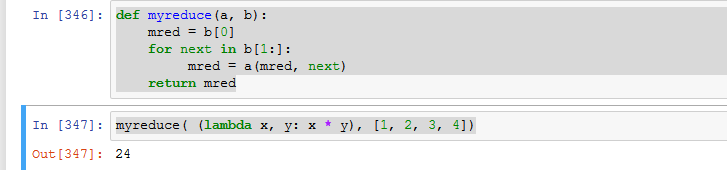
mred = b[0]

for next in b[1:]:

mred = a(mred, next)

return mred

myreduce( (lambda x, y: x \* y), [1, 2, 3, 4])



# Q1.2: Write a Python program to implement your own myfilter() function which works exactly like Python's built-in function filter()

# Answer :

alist = [1,2,3,4,5]

def myfilter(x):

if (x%2)== 0 :

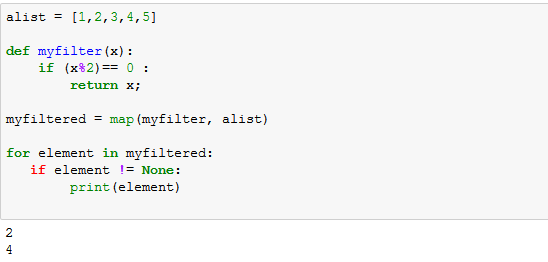
return x;

myfiltered = map(myfilter, alist)

for element in myfiltered:

if element != None:

print(element)



# Q2: Implement List comprehensions to produce the following lists.

# Write List comprehensions to produce the following Lists

['A', 'C', 'A', 'D', 'G', 'I', ’L’, ‘ D’]

['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']

['x', 'y', 'z', 'xx', 'yy', 'zz', 'xx', 'yy', 'zz', 'xxxx', 'yyyy', 'zzzz']

[[2], [3], [4], [3], [4], [5], [4], [5], [6]]

[[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]

[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]

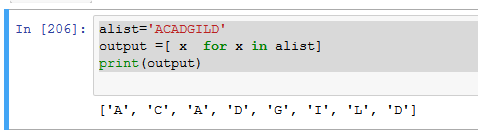
# Answer :

1. ['A', 'C', 'A', 'D', 'G', 'I', ’L’, ‘ D’]

alist='ACADGILD'

output =[ x for x in alist]

print(output)

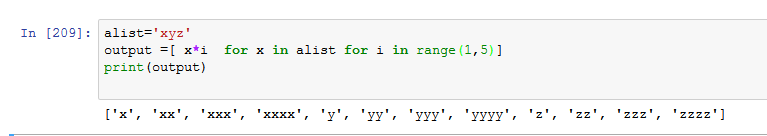


1. ['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']

alist='xyz'

output =[ x\*i for x in alist for i in range(1,5)]

print(output)

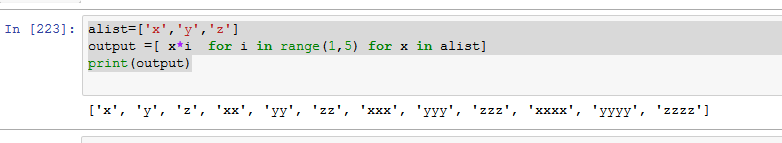


1. ['x', 'y', 'z', 'xx', 'yy', 'zz', 'xxx’, 'yyy', 'zzz', 'xxxx', 'yyyy', 'zzzz']

alist=['x','y','z']

output =[ x\*i for i in range(1,5) for x in alist]

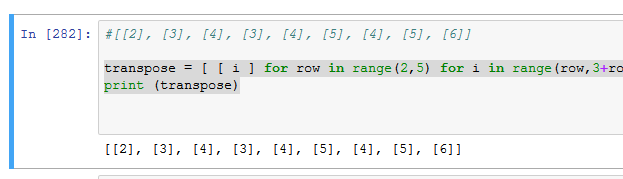
print(output)



1. [[2], [3], [4], [3], [4], [5], [4], [5], [6]]

transpose = [ [ i ] for row in range(2,5) for i in range(row,3+row)]

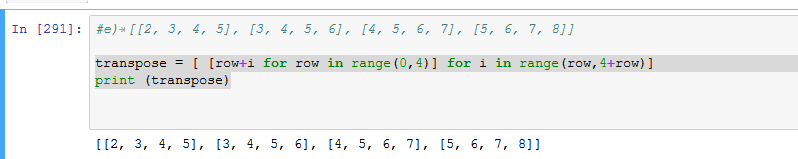
print (transpose)



1. [[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]

transpose = [ [row+i for row in range(0,4)] for i in range(row,4+row)]

print (transpose)

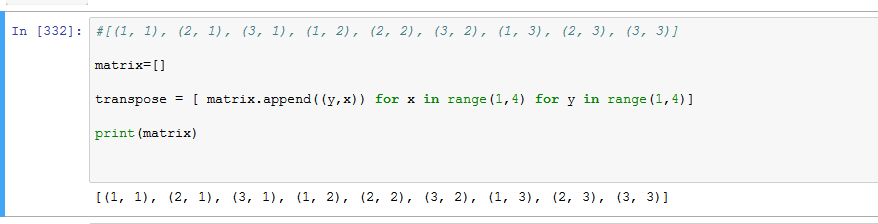


1. [(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]

matrix=[]

transpose = [ matrix.append((y,x)) for x in range(1,4) for y in range(1,4)]

print(matrix)



# Q3: Implement a function longestWord() that takes a list of words and returns the longest

# one.

# Answer :

alist=['Java Programming','Unix program', 'shell programming', 'data science']

def highest\_word(alist):

longestWord = ""

max\_len = 0

for word in alist:

if len(word) > max\_len:

max\_len = len(word)

longestWord = word

return longestWord

output =(highest\_word(alist))

print(output)

