Session 3, Assignment 3

# Question 1:

Write a Python Program to implement your own reduce() function which works exactly like Python's built-in function reduce()

# Answer:

def myreduce(anyfunc, sequence):

result = sequence[0]

for item in sequence[1:]:

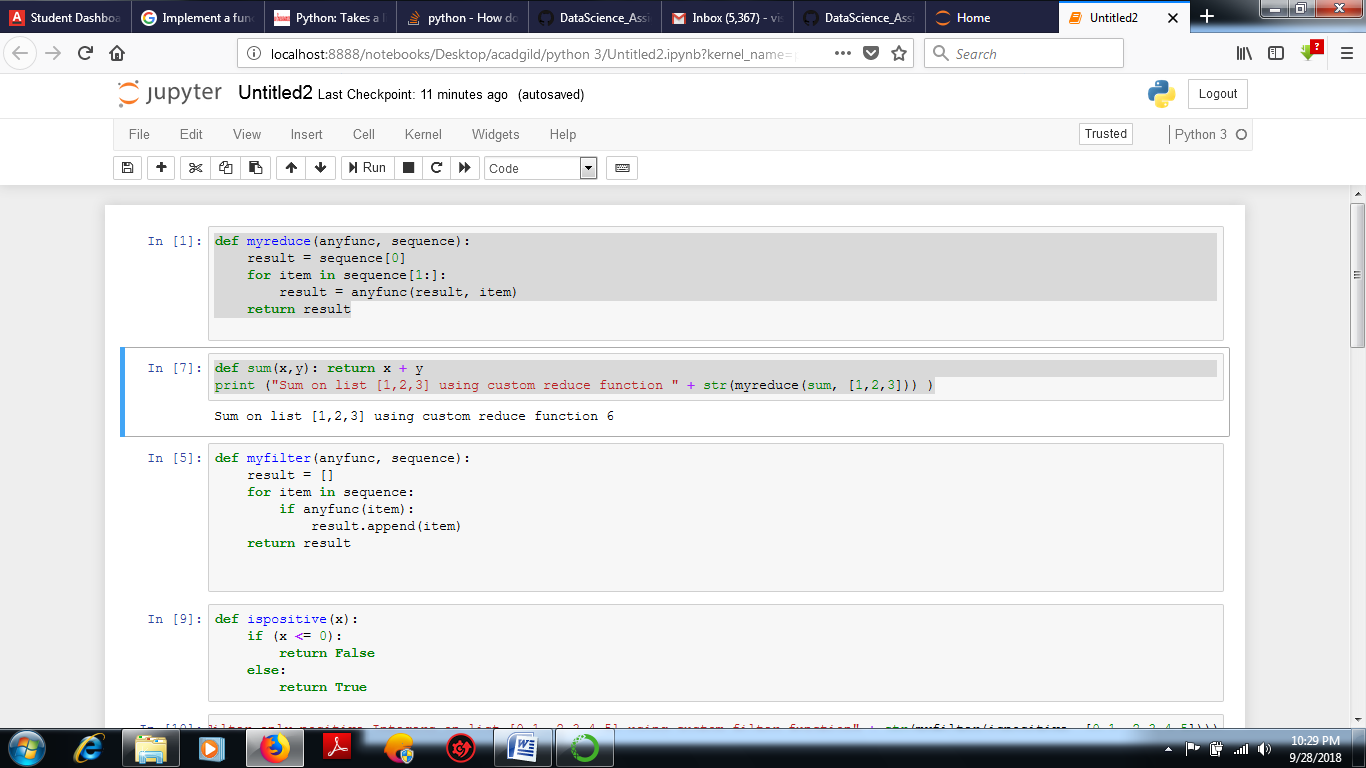
result = anyfunc(result, item)

return result

def sum(x,y): return x + y

print ("Sum on list [1,2,3] using custom reduce function " + str(myreduce(sum, [1,2,3])) )

Result: Sum on list [1,2,3] using custom reduce function 6



# Question 2:

Write a Python program to implement your own myfilter() function which works exactly like

Python's built-in function filter()

# Answer:

def myfilter(anyfunc, sequence):

result = []

for item in sequence:

if anyfunc(item):

result.append(item)

return result

def ispositive(x):

if (x <= 0):

return False

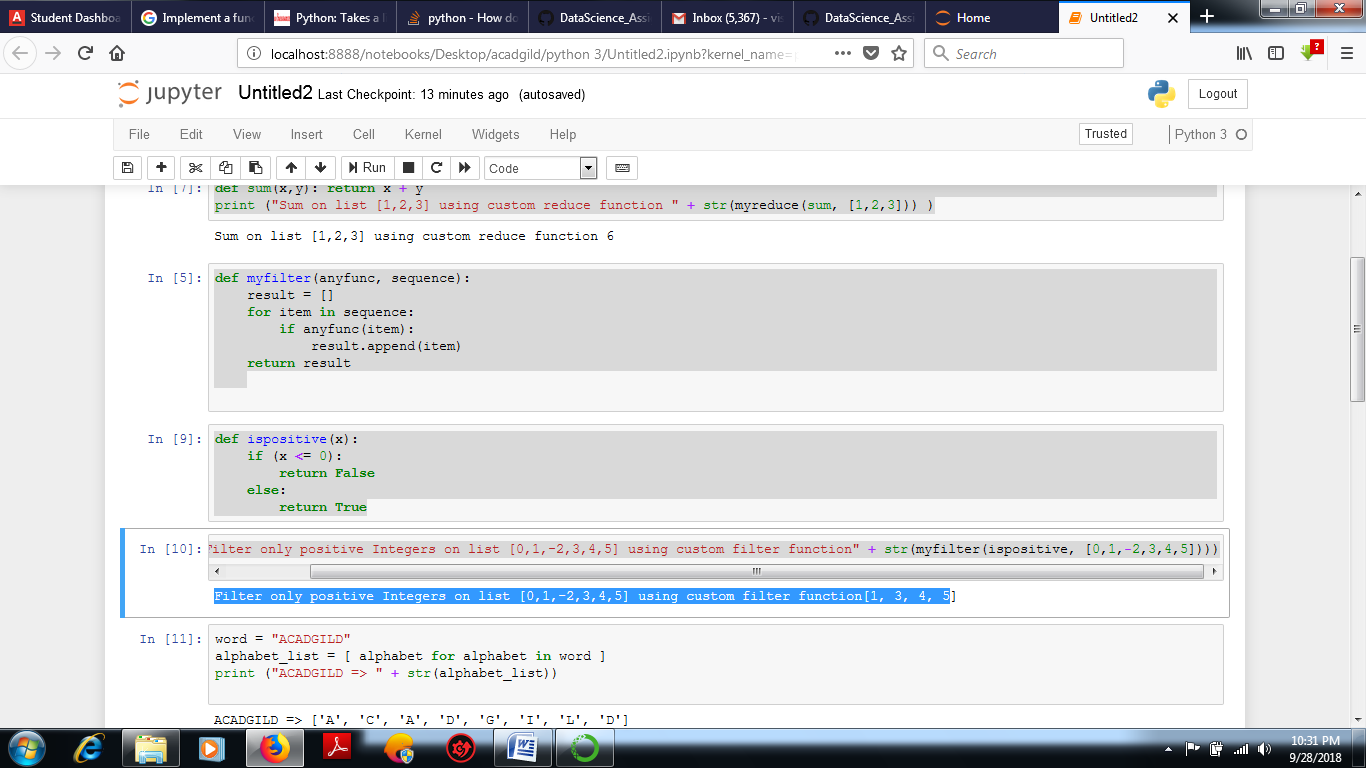
else:

return True

print ("Filter only positive Integers on list [0,1,-2,3,4,5] using custom filter function" + str(myfilter(ispositive, [0,1,-2,3,4,5])))

# # Result

Filter only positive Integers on list [0,1,-2,3,4,5] using custom filter function[1, 3, 4, 5



# Question 3:

|  |
| --- |
| 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', 'xxx', 'yyy', 'zzz', '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:

word = "ACADGILD"

alphabet\_list = [ alphabet for alphabet in word ]

print ("ACADGILD => " + str(alphabet\_list))

input\_list = ['x','y','z']

result = [ item\*num for item in input\_list for num in range(1,5) ]

print("['x','y','z'] => " + str(result))

input\_list = ['x','y','z']

result = [ item\*num for num in range(1,5) for item in input\_list ]

print("['x','y','z'] => " + str(result))

input\_list = [2,3,4]

result = [ [item+num] for item in input\_list for num in range(0,3)]

print("[2,3,4] =>" + str(result))

input\_list = [2,3,4,5]

result = [ [item+num for item in input\_list] for num in range(0,4) ]

print("[2,3,4,5] =>" + str(result))

input\_list=[1,2,3]

result = [ (b,a) for a in input\_list for b in input\_list]

print("[1,2,3] =>" + str(result))

# Result:

ACADGILD => ['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']

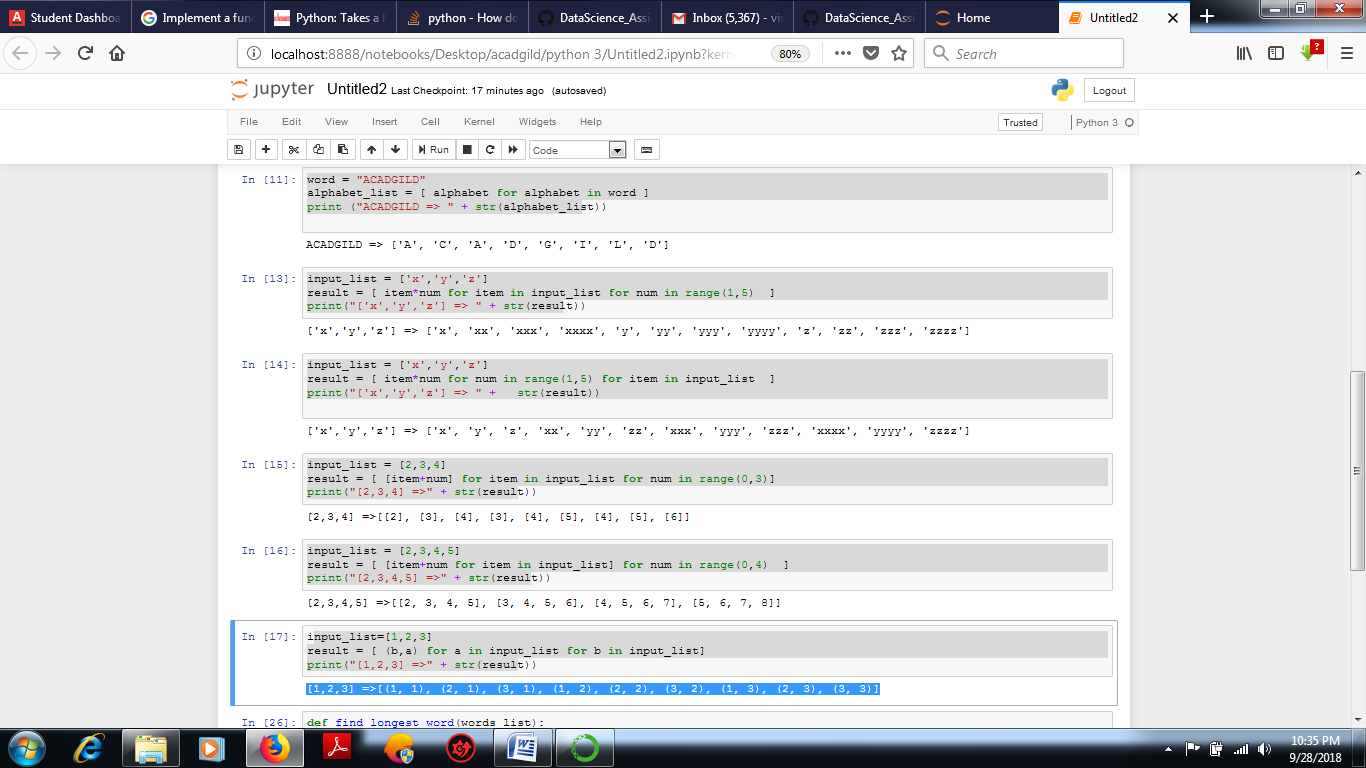
['x','y','z'] => ['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']

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

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

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

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



# Question 4:

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

# Answer:

# def find\_longest\_word(words\_list):

# word\_len = []

# for n in words\_list:

# word\_len.append((len(n), n))

# word\_len.sort()

# return word\_len[-1][1]

# print(find\_longest\_word(["Acadgild", "VishalSoni", "DataScience"]))

# #Result:

DataScience

