1.1 Write a Python Program to implement your own myreduce() function which works exactly

like Python's built-in function reduce()

Code:

def my\_reduce(func, seq):

first = seq[0]

for i in seq[1:]:

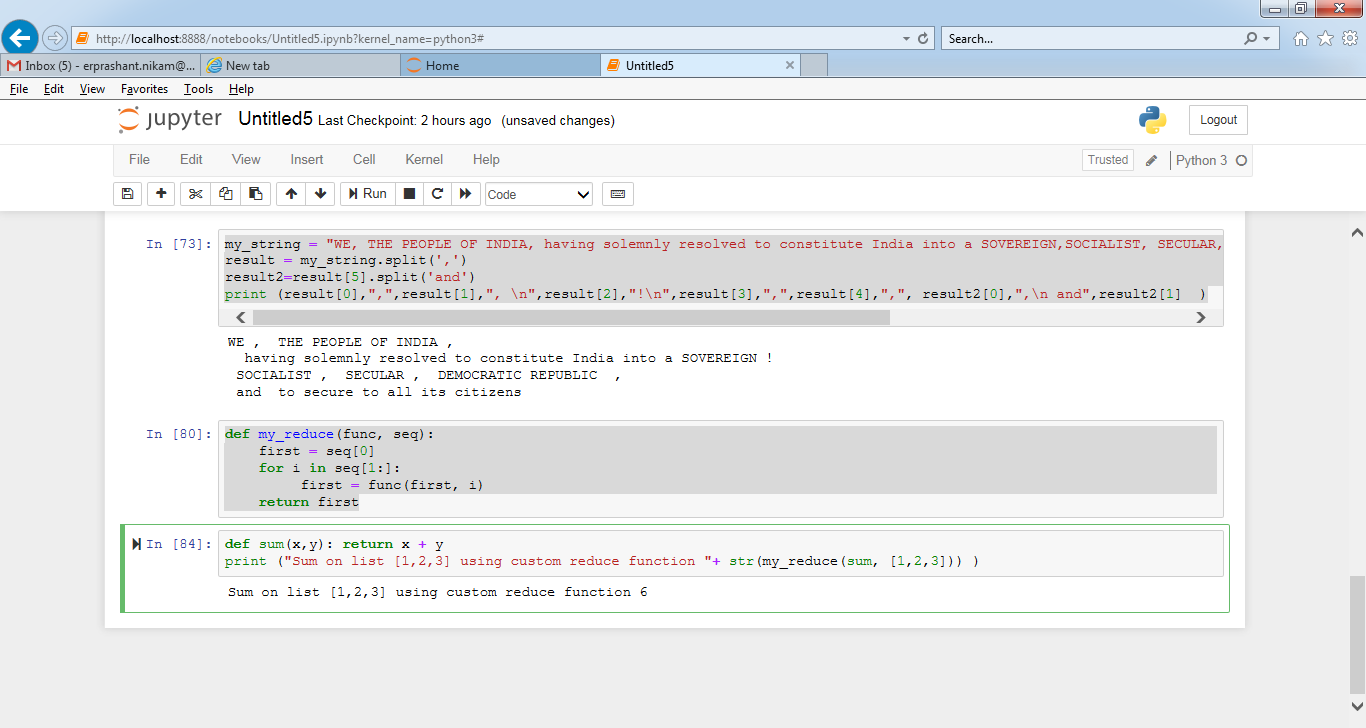
first = func(first, i)

return first

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

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

Output :



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

Python's built-in function filter()

Code:

def myfilter(anyfunc, sequence):

result = []

for item in sequence:

if anyfunc(item):

result.append(item)

return result

def ispositive(x):

if (x <= 0):

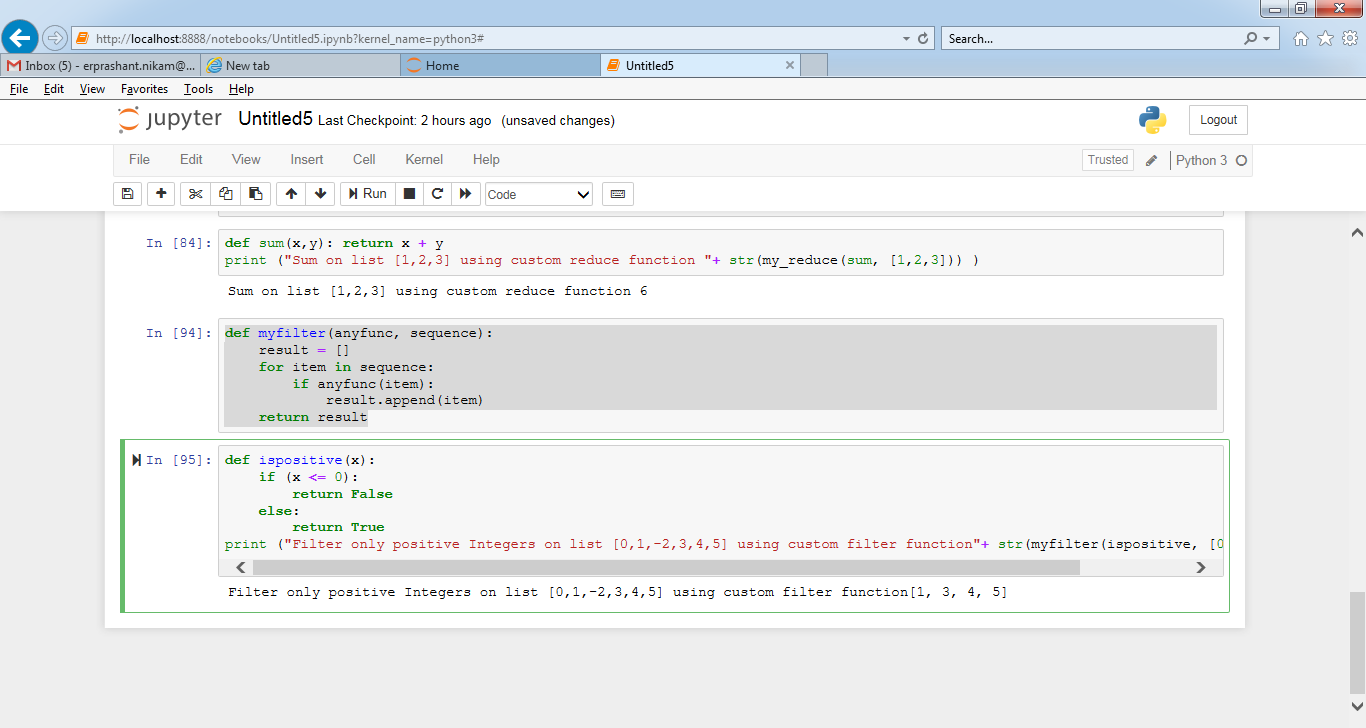
return False

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])))

OutPut:



2. Implement List comprehensions to produce the following lists.

Write List comprehensions to produce the following Lists

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

word = "ACADGILD"

alphabet\_list = [ alphabet for alphabet in word ]

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

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

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))

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

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))

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

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))

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

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))

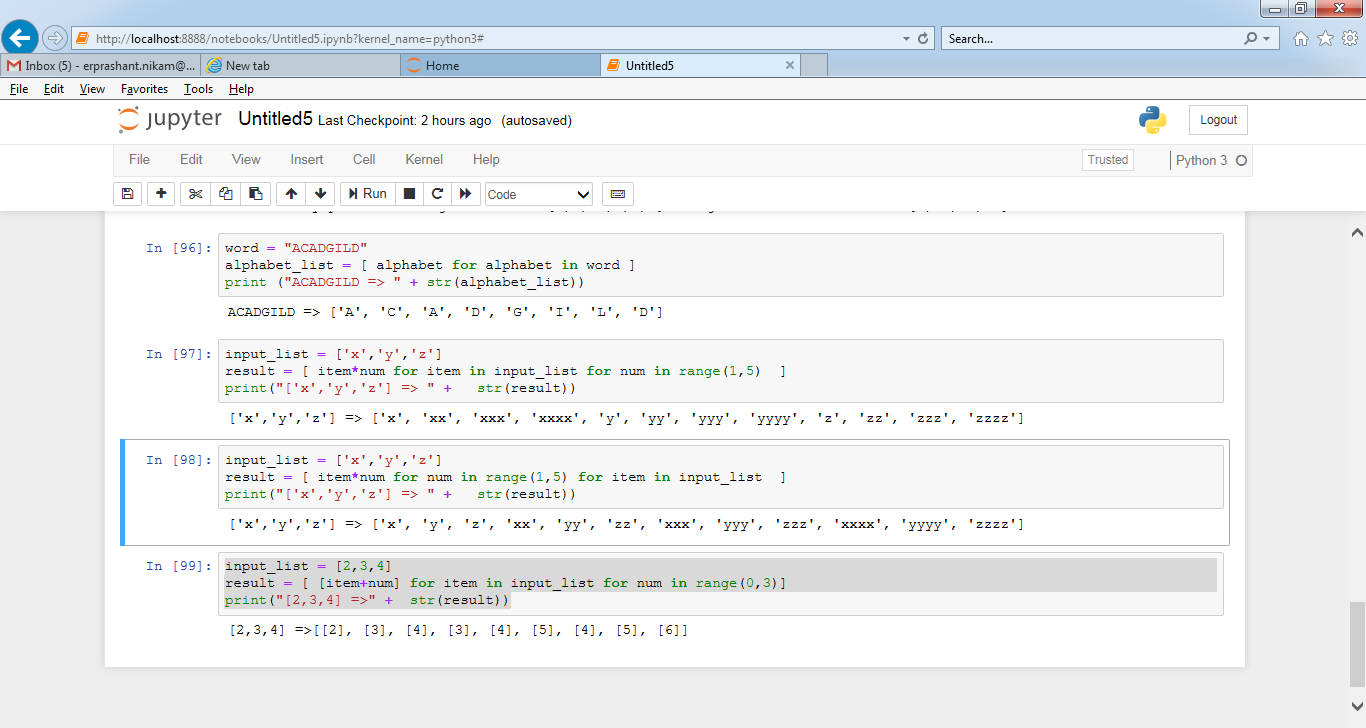
[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]

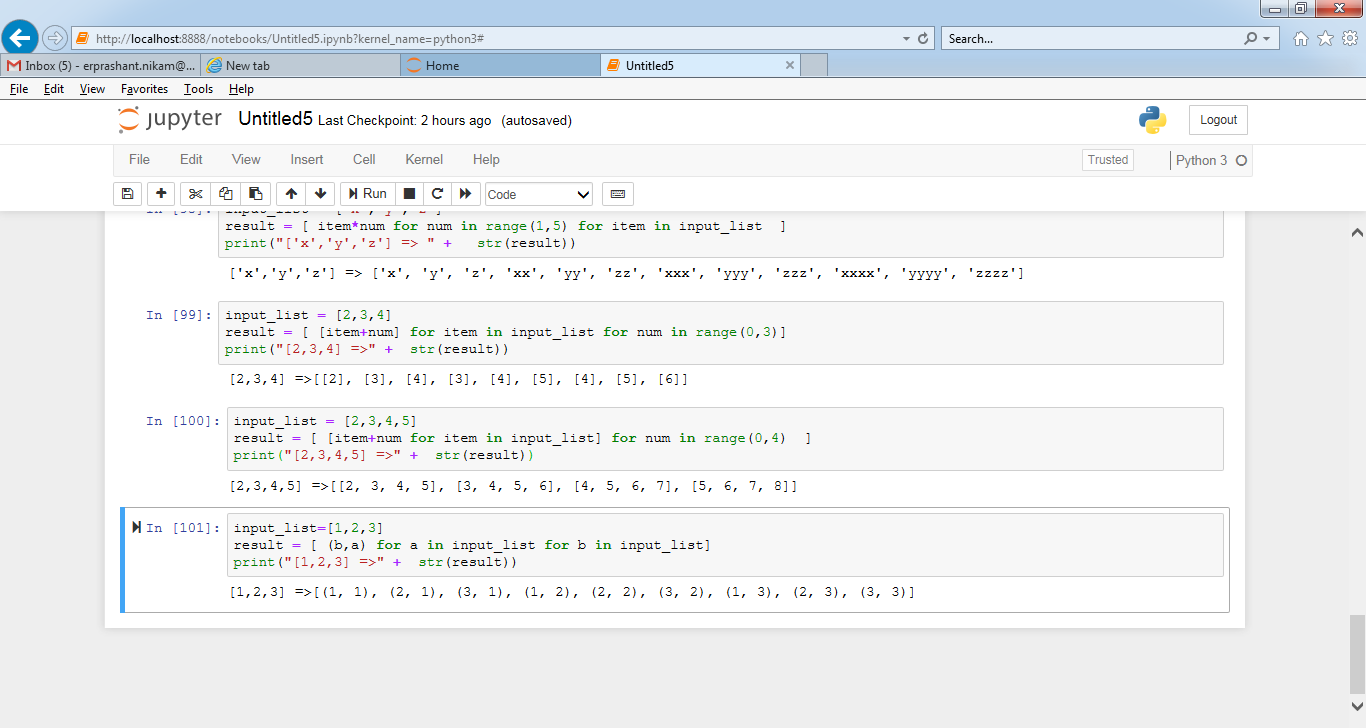
input\_list=[1,2,3]

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

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

OutPut:





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

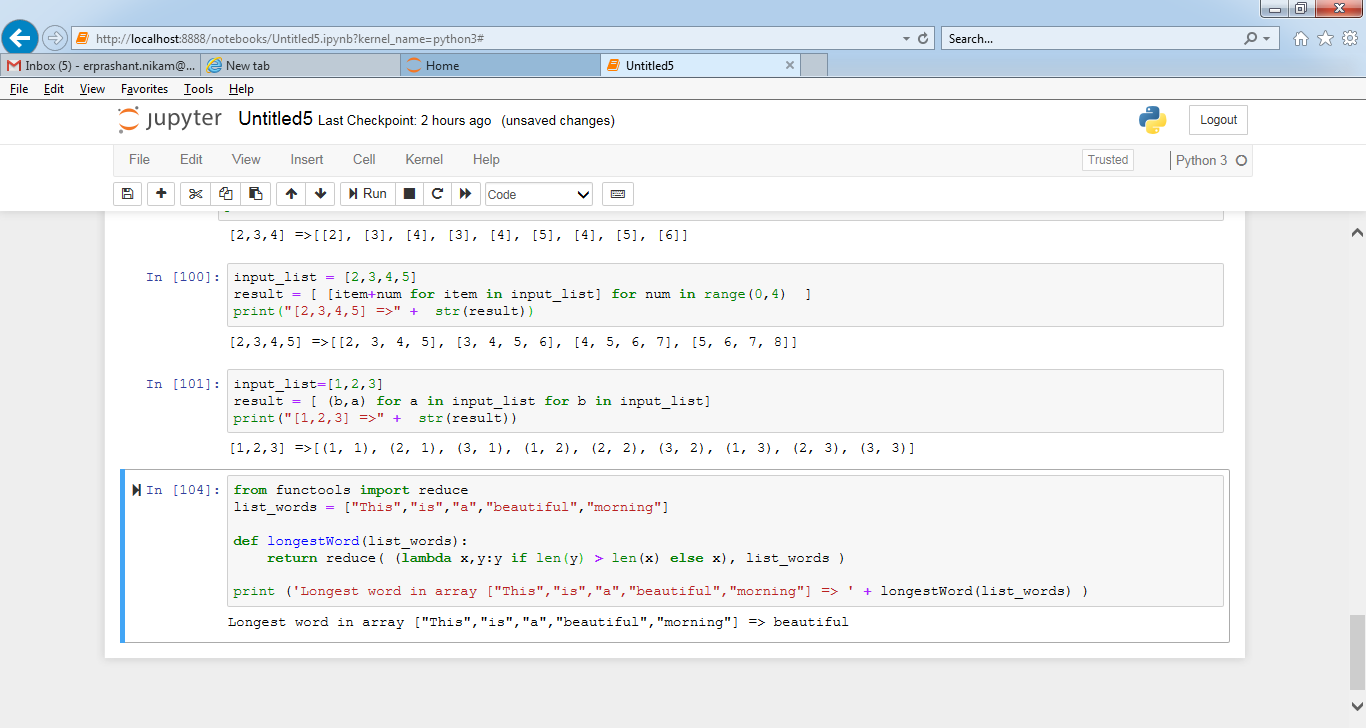
from functools import reduce

list\_words = ["This","is","a","beautiful","morning"]

def longestWord(list\_words):

return reduce( (lambda x,y:y if len(y) > len(x) else x), list\_words )

print ('Longest word in array ["This","is","a","beautiful","morning"] => ' + longestWord(list\_words) )



NOTE:​ ​The​ ​solution​ ​shared​ ​through​ ​Github​ ​should​ ​contain​ ​the​ ​source code​ ​used​ ​and​ ​the​

​screenshot​ ​of​ ​the​ ​output.