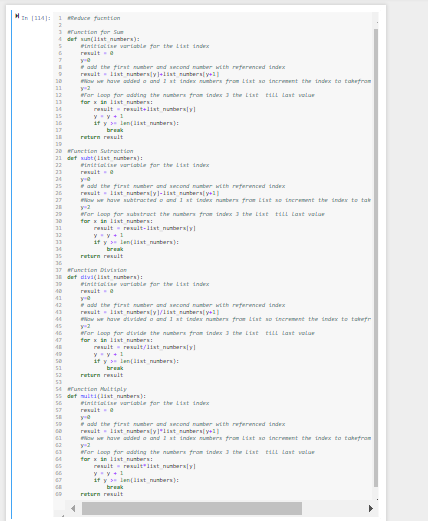
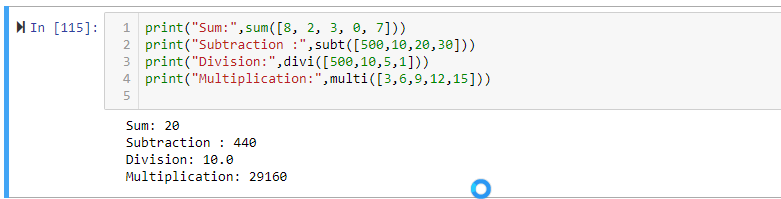
* 1. Write a Python Program to implement your own myreduce() function which works exactly like Python's built-in function reduce()

Code Screen Shot:



**Result:**



**Code:**

#Reduce function

#Function for Sum

def sum(list\_numbers):

#initialise variable for the list index

result = 0

y=0

# add the first number and second number with referenced index

result = list\_numbers[y]+list\_numbers[y+1]

#Now we have added o and 1 st index numbers from list so increment the index to takefrom the 3 value for loop

y=2

#For loop for adding the numbers from index 3 the list till last value

for x in list\_numbers:

result = result+list\_numbers[y]

y = y + 1

if y >= len(list\_numbers):

break

return result

#Function Sutraction

def subt(list\_numbers):

#initialise variable for the list index

result = 0

y=0

# add the first number and second number with referenced index

result = list\_numbers[y]-list\_numbers[y+1]

#Now we have subtracted o and 1 st index numbers from list so increment the index to takefrom the 3 value for loop

y=2

#For loop for substract the numbers from index 3 the list till last value

for x in list\_numbers:

result = result-list\_numbers[y]

y = y + 1

if y >= len(list\_numbers):

break

return result

#Function Division

def divi(list\_numbers):

#initialise variable for the list index

result = 0

y=0

# add the first number and second number with referenced index

result = list\_numbers[y]/list\_numbers[y+1]

#Now we have divided o and 1 st index numbers from list so increment the index to takefrom the 3 value for loop

y=2

#For loop for divide the numbers from index 3 the list till last value

for x in list\_numbers:

result = result/list\_numbers[y]

y = y + 1

if y >= len(list\_numbers):

break

return result

#Function Multiply

def multi(list\_numbers):

#initialise variable for the list index

result = 0

y=0

# add the first number and second number with referenced index

result = list\_numbers[y]\*list\_numbers[y+1]

#Now we have added o and 1 st index numbers from list so increment the index to takefrom the 3 value for loop

y=2

#For loop for adding the numbers from index 3 the list till last value

for x in list\_numbers:

result = result\*list\_numbers[y]

y = y + 1

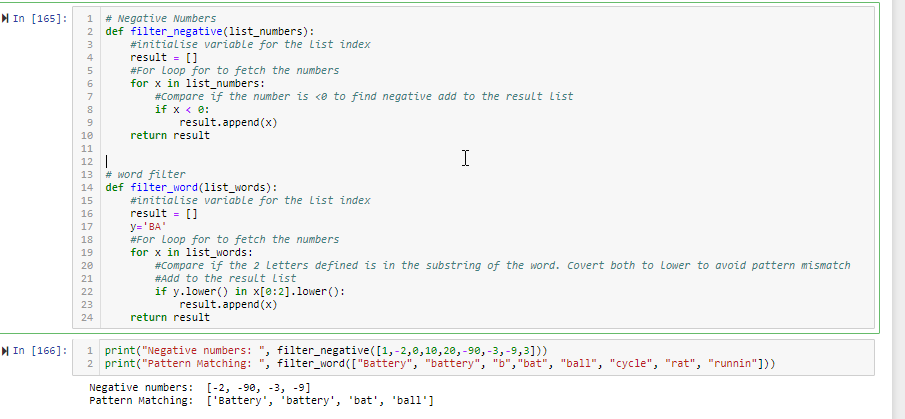
if y >= len(list\_numbers):

break

return result

1.2 Write a Python program to implement your own myfilter() function which works exactly like Python's built-in function filter()

**Result:**



**Code:**

# Negative Numbers

def filter\_negative(list\_numbers):

#initialise variable for the list index

result = []

#For loop for to fetch the numbers

for x in list\_numbers:

#Compare if the number is <0 to find negative add to the result list

if x < 0:

result.append(x)

return result

# word filter

def filter\_word(list\_words):

#initialise variable for the list index

result = []

y='BA'

#For loop for to fetch the numbers

for x in list\_words:

#Compare if the 2 letters defined is in the substring of the word. Covert both to lower to avoid pattern mismatch

#Add to the result list

if y.lower() in x[0:2].lower():

result.append(x)

return result

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’]

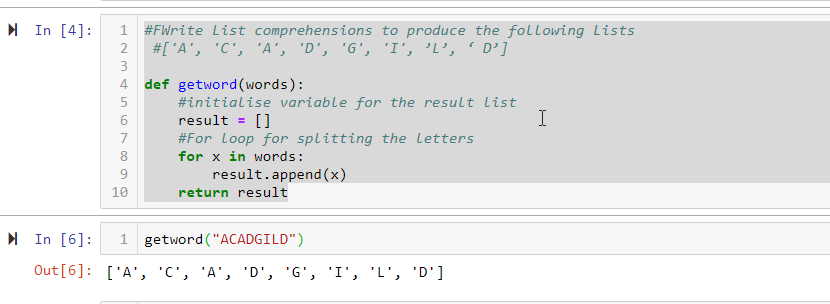
['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']

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

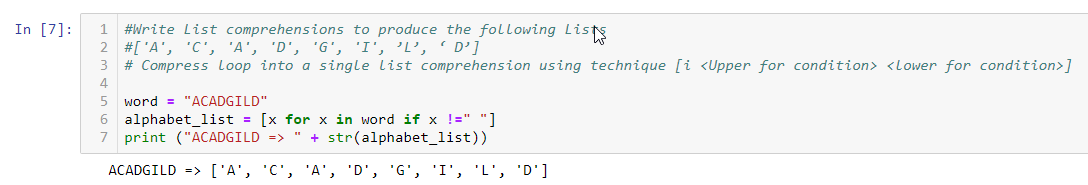
[[2], [3], [4], [3], [4], [5], [4], [5], [6]]

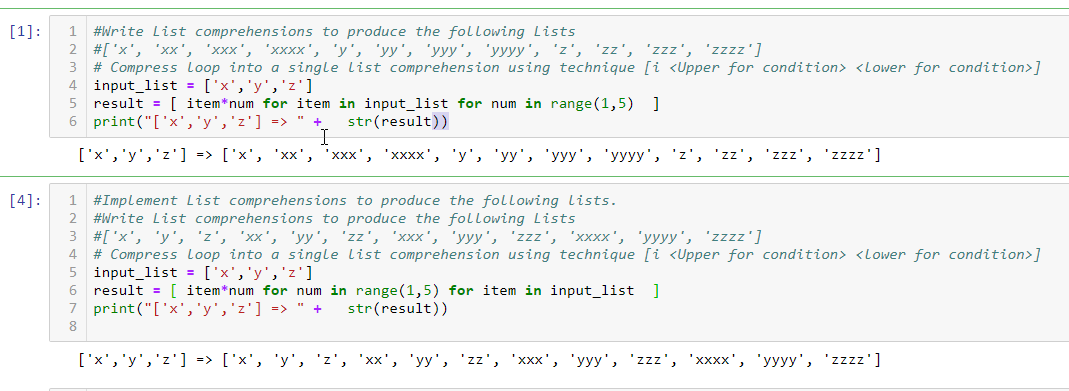
[[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]

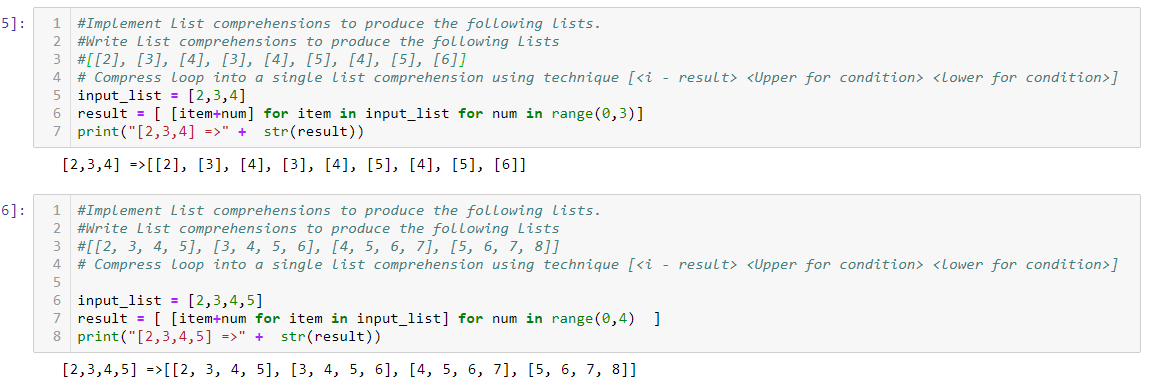
Result:

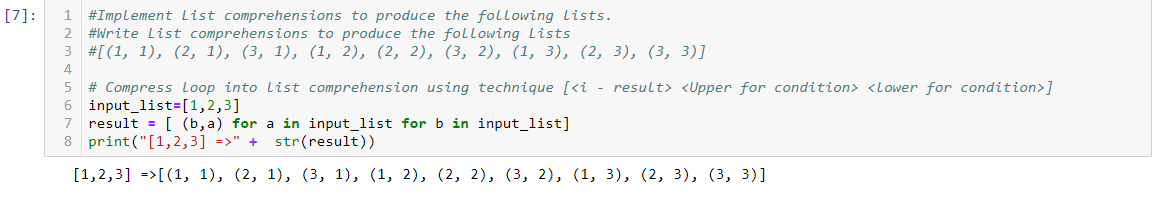


Comprehensions method









Code:

#Write List comprehensions to produce the following Lists

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

#Function to get th words -Normal loop method

def getword(words):

#initialise variable for the result list

result = []

#For loop for splitting the letters

for x in words:

result.append(x)

return result

#Write List comprehensions to produce the following Lists

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

# Compress loop into a single list comprehension using technique [i <Upper for condition> <lower for condition>]

word = "ACADGILD"

alphabet\_list = [x for x in word if x !=" "]

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

#Write List comprehensions to produce the following Lists

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

# Compress loop into a single list comprehension using technique [i <Upper for condition> <lower for condition>]

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))

#Implement List comprehensions to produce the following lists.

#Write List comprehensions to produce the following Lists

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

# Compress loop into a single list comprehension using technique [i <Upper for condition> <lower for condition>]

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))

#Implement List comprehensions to produce the following lists.

#Write List comprehensions to produce the following Lists

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

# Compress loop into a single list comprehension using technique [<i - result> <Upper for condition> <lower for condition>]

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))

#Implement List comprehensions to produce the following lists.

#Write List comprehensions to produce the following Lists

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

# Compress loop into a single list comprehension using technique [<i - result> <Upper for condition> <lower for condition>]

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))

#Implement List comprehensions to produce the following lists.

#Write List comprehensions to produce the following Lists

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

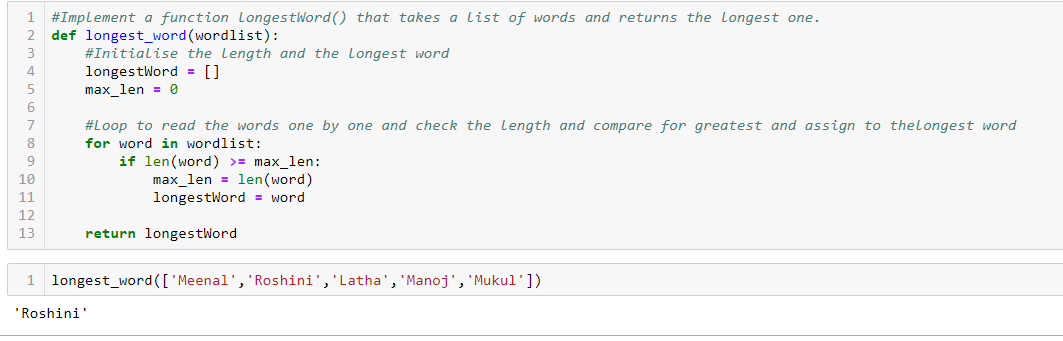
# Compress loop into list comprehension using technique [<i - result> <Upper for condition> <lower for condition>]

input\_list=[1,2,3]

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

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

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



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

def longest\_word(wordlist):

#Initialise the length and the longest word

longestWord = []

max\_len = 0

#Loop to read the words one by one and check the length and compare for greatest and assign to thelongest word

for word in wordlist:

if len(word) >= max\_len:

max\_len = len(word)

longestWord = word

return longestWord