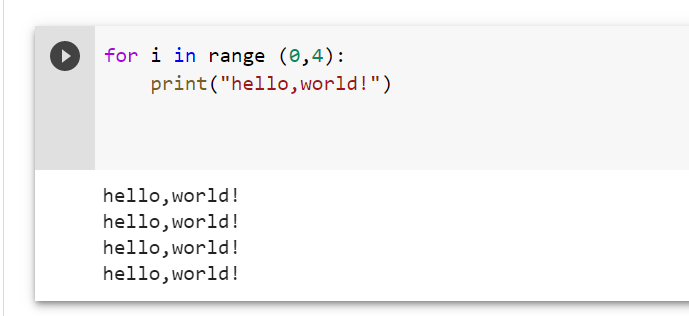
**LAB 2**

**OPEN SOURCE**

Q1.

for i in range (0,4):

print("hello,world!")



Q2.

def duplicate(lst):

lst1 = []

for i in lst:

if i not in lst1:

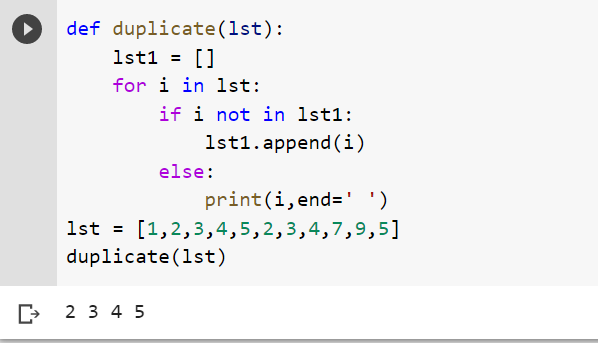
lst1.append(i)

else:

print(i,end=' ')

lst = [1,2,3,4,5,2,3,4,7,9,5]

duplicate(lst)



Q3.

my\_list = ['geeks', 'for', 'geeks', 'like',

'geeky','nerdy', 'geek', 'love',

'questions','words', 'life']

def divide\_chunks(l, n):

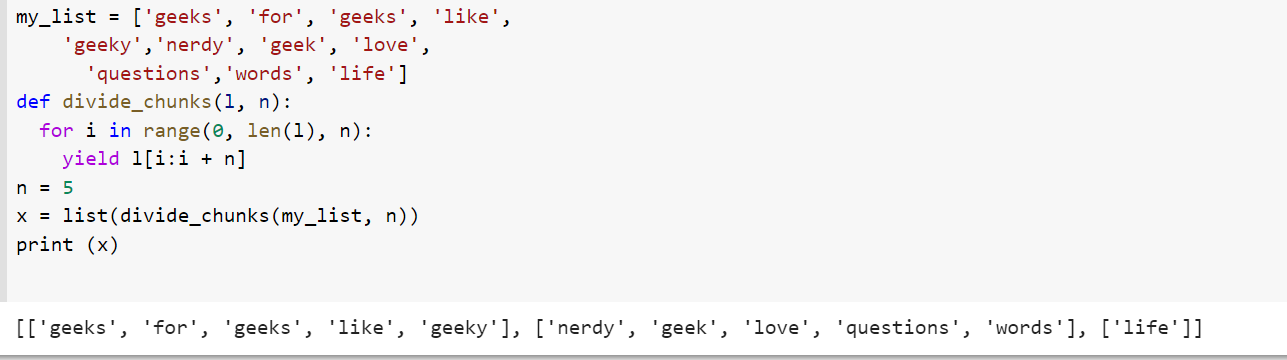
for i in range(0, len(l), n):

yield l[i:i + n]

n = 5

x = list(divide\_chunks(my\_list, n))

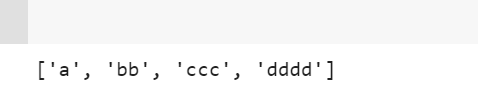
print (x)



Q4

xs = ['dddd','a','bb','ccc']

sorted(xs, key=len)



Q5.

def Sorting(lst):

lst2 = sorted(lst, key=len)

return lst2

lst = ["rohan", "amy", "sapna", "muhammad",

"aakash", "raunak", "chinmoy"]

print(Sorting(lst))



#Q6

f=open("new.txt","w+")

for i in range(10):

f.write("This is sample text and line no %d\r\n" %(i+1))

f.close()

f = open("new.txt","r")

if f.mode == 'r':

contents = f.read()

print (contents)

#Q7

file = open("sample.txt","r")

number\_of\_lines = 0

number\_of\_words = 0

number\_of\_characters = 0

for line in file:

line = line.strip("\n")

words = line.split()

number\_of\_lines += 1

number\_of\_words += len(words)

number\_of\_characters += len(line)

file.close()

print("lines:",number\_of\_lines,"words:",number\_of\_words,"characters:",number\_of\_characters)

#Q8

textfile = open("sample.txt")

lines = textfile.readlines()

for line in reversed(lines):

print(line)

textfile.close()

#Q9

def revline(x):

i=0

z=len(open(x).readlines())

rev=[None]\*z

f=open(x)

while(i<z):

rev[i]=f.readline()

rev[i]=rev[i].strip()

print(rev[i][::-1])

i=i+1

revline("sample.txt")

#Q10

def wrap(filename,k):

f=open(filename).readlines()

for i in f:

new = i

while len(new)>k:

print(new[:k])

new=new[k:]

print(new)

wrap("sample.txt",5)

#Q11

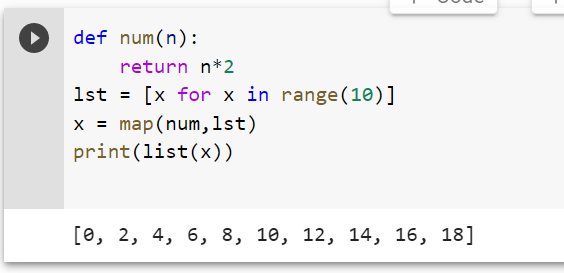
def num(n):

return n\*2

lst = [x for x in range(10)]

x = map(num,lst)

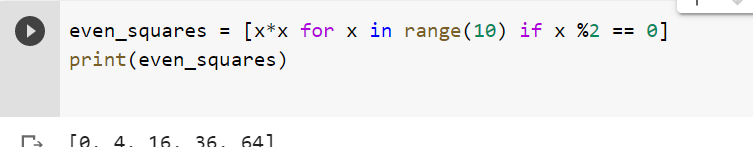
print(list(x))



#Q12

even\_squares = [x\*x for x in range(10) if x %2 == 0]

print(even\_squares)

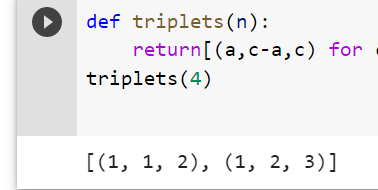


#Q13

def triplets(n):

return[(a,c-a,c) for c in range(2,n) for a in range(1,c//2+1)]

triplets(4)



#Q14

import csv

with open('employee\_birthday.txt', mode='r') as csv\_file:

csv\_reader = csv.DictReader(csv\_file)

line\_count = 0

for row in csv\_reader:

if line\_count == 0:

print(f'Column names are {", ".join(row)}')

line\_count += 1

print(f'\t{row["name"]} works in the {row["department"]} department, and was born in {row["birthday month"]}.')

line\_count += 1

print(f'Processed {line\_count} lines.')

#Q15

import csv

with open('employee\_birthday.txt') as csv\_file:

csv\_reader = csv.reader(csv\_file, delimiter=',')

line\_count = 0

for row in csv\_reader:

if line\_count == 0:

print(f'Column names are {", ".join(row)}')

line\_count += 1

else:

print(f'\t{row[0]} works in the {row[1]} department, and was born in {row[2]}.')

line\_count += 1

print(f'Processed {line\_count} lines.')

#Q16

def mutate(d):

ret=[d]

i=0

l=len(d)

alp=map(chr,range(97,123))

while i<l:

cop=d

ret.append(cop[:i]+cop[i+1:])

if i<=l-2:

ret.append(cop[:i]+cop[i+1]+cop[i]+cop[i+2:])

elif i<l-1:

ret.append(cop[:i]+cop[i+1]+cop[i])

for x in alp:

ret.append(cop[:i]+x+cop[i+1:])

for x in alp:

ret.append(d+x)

ret.append(x+d)

ret.append(cop[:i]+x+cop[i:])

i=i+1

return ret

print ('hefllo') in mutate('hello')

print ('hllo') in mutate('hello')



#Q17

def nearly\_equal(str1,str2):

count=0

i=0

j=0

while(i<len(str1) and j<len(str2)):

if(str1[i]!=str2[j]):

count=count+1

if(len(str1)>len(str2)):

i=i+1

elif(len(str1)==len(str2)):

pass

else:

i=i-1

if(count>1):

return False

i=i+1

j=j+1

if(count<2):

return True

str1=input("Enter first string: ")

str2=input("Enter second string: ")

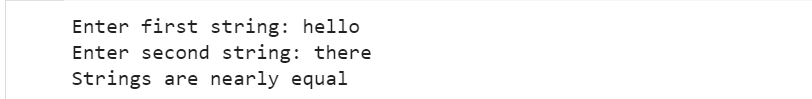
boolean=nearly\_equal(str1,str2)

if(boolean):

print("Strings are nearly equal")

else:

print("Strings are not equal")



#Q18

f="sample.txt"

file= open(f,"r")

a=[]

b={}

for i in file:

for j in range(0,len(i)):

a.append(i[j])

for i in a:

if i in b:

b[i]+=1

else:

b[i]=1

print(b)

c=f.split(".")

if c[1]=="txt":

print("\n\nIt is a text file")

elif c[1]=="cpp":

print("\n\nIt is a C++ file")

else:

print("\n\nIt is a C file")

#19

def anagrams(x):

from itertools import permutations

s={}

while len(x)>0:

x1=x.pop()

s[x1]=s.get(x1,[])

s[x1].append(x1)

i=0

while i<len(x):

z1=x[i]

perm=[''.join(p) for p in permutations(x1)]

if z1 in perm:

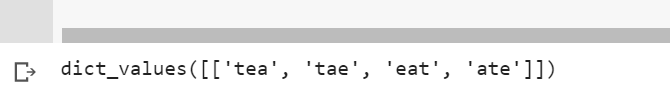
x.remove(z1)

s[x1].append(z1)

else:i=i+1

return s.values()

print(anagrams(['tae','souep','eat','ihba','node','peuos','ate','abhi','bhia','done','soupe','tea']))



Q21

def areAnagram(str1, str2):

# Get lengths of both strings

n1 = len(str1)

n2 = len(str2)

# If lenght of both strings is not same, then

# they cannot be anagram

if n1 != n2:

return 0

# Sort both strings

str1 = sorted(str1)

str2 = sorted(str2)

# Compare sorted strings

for i in range(0, n1):

if str1[i] != str2[i]:

return 0

return 1

# Driver code

str1 = "test"

str2 = "ttew"

# Function Call

if areAnagram(str1, str2):

print("The two strings are anagram of each other")

else:

print("The two strings are not anagram of each other")

