for fizzbuzz in range(1, 100):

if fizzbuzz % 15 == 0:

print("fizzbuzz")

continue

elif fizzbuzz % 3 == 0:

print("fizz")

continue

elif fizzbuzz % 5 == 0:

print("buzz")

continue

print(fizzbuzz)

total = int(input("enter the total user"))

staff\_user=int(input("enter the staff user"))

non\_teaching=staff\_users/3

student\_user=total-(staff\_user+non\_teaching)

n=int(student\_users)

print(n)

class solution:

def smallernumbersthancurrent(self, numns: list[int]) -> list[int]:

result = []

for num in nums:

count = 0

for comp in nums:

if num > comp:

count += 1

result.append(count)

return result

ob = solution()

print(ob.smallernumbersthancurrent([6,5,4,8,]))

class solution(object):

def ispalidrome(self, s):

type s: str

rtype: bool

x = ""

diff = ord('a') - ord('A')

for i in s:

if ord(i)>=('a') and ord(i)<=ord('z') or ord(i)>=ord("0") and(i)<+ord("9"):

x+=i

elif ord(I)>=ord('A') and ord (I)<=ord('z'):

i = chr(diff+ord(i))

x+=i

return x ==x[::-1]

ob1 = solution()

print(ob1.ispalidrome("a man, a paln, a canal: panam"))

def minjumps(arr,n):

if(n <= 1):

return 0

if(arr[0] == 0):

return - 1

maxreach = arr[0]

step =arr[0]

jump = 1

for i in range(1,n):

if(i == n -1):

return jump

maxrreach = max(maxreach, i +arr[i])

step -=1;

if(step==0):

jump += 1

if(i>=maxreach):

return - 1

step = maxreach - i;

return - 1

arr = [1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9]

size = len(arr)

print(minjumps(arr,size))

def removechar(s, c):

count = s.count(c)

s = list(s)

while counts:

s.remove(c)

count -=1

s = ''.join(s)

print(s)

if\_name\_=='\_main\_':

s = "hello world"

removechar(s, '1')

def countstrings(n, start):

if n == 0:

return 1

count = 0

for i in range(start, 5):

count += countstring(n - 1,i)

return count

def countvowelstring(n):

return countstrings(n, 0)

n = 1

print(countvowelstring(n))

def value(r):

if ( r == 'i' ):

return 1

if ( r == 'v' ):

return 5

if ( r == 'x'):

return 10

if ( r == 'l'):

return 50

if ( r == 'c'):

return 100

if ( r == 'd' ):

return 500

if ( r == 'm'):

return 1000

return -1

def romantodecimal(str):

res = 0

i = 0

while (i < len(str)):

s1 = value(str[i])

if (i + 1 < len(str)):

s2 = value(str[i + 1])

if (s1 >= s2):

res = res + s1

i = i + 1

else:

res = res + s2 - s1

i = i + 2

else:

res = res + s1

i = i + 1

return res

print("integer from of roman numeral is"),

print(romanto decimal("lvii"))

month = input("input the month:")

day = int (input("enter the day:"))

if month in ('january','febuary','march'):

season = 'winter'

elif month in ('april', 'may' , 'june'):

season = 'summer'

elif month in ('july', 'august',' september'):

season = 'spring'

else:

season = 'autumn'

if (month == 'june' ) and (day > 20) :

season = 'spring'

elif (month == 'march' ) and ( day > 19):

season = 'summer'

elif (month == 'september') and (day > 21):

season = 'autum'

elif ( month == ' december' ) and ( day > 20 ):

season = 'winter'

print("season is", season)

from random import sample

test\_list = [,'python', 'program', 'are', 'very', 'difficilut']

print("the original list : " + str(test\_list))

res = [''.join(sample(ele, len(ele)) for ele in test\_list]

print("scrambled string in list are : " + str(res))