1.Write a python program to right rotate a List by n Enter position to rotate list item: 3 Sample input: [10, 20, 30, 40, 50, 60, 70] Expected output: [50, 60, 70, 10, 20, 30, 40]

-------------------------------------------------------------------

l=[10,20,30,40,50,60,70]

n=int(input())

for i in range(n):

    x=l[0]

    for j in range(len(l)-1):

        l[j]=l[j+1]

    l[-1]=x

print(l)

1. Write a python program where for every two hours it prints the pattern without using sleep function

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*

---------------------------------------------------------------------------------------------------------------------------

import math

import schedule

def pattern():

    print(

    '''\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*''')

schedule.every(2).hours.do(pattern)

while 1:

    schedule.run\_pending()

---------------------------------------------------------------------------------------------------------------------------

3.Write a program using decorators to print the traffic signal messages Expected output - RED : STOP YELLOW : SLOW DOWN GREEN : GO The decorator should be working in this order

---------------------------------------------------------------------------------------------------------

def trafficcolor(func):

    def message():

        color=input("Enter the color of traffic signal:").lower()

        if color in ("red","yellow","green"):

            func(color)

        else:

            print("invalid color")

    return message

@trafficcolor

def trafficsignal(color):

    if color=="red":

        print("STOP")

    elif color=="yellow":

        print("SLOW DOWN")

    else:

        print("GO")

trafficsignal()

----------------------------------------------------------------------------------------------------------

4.How do you open a file of large size, say around 10GB? So that program should not crash

-----------------------------------------------------------------------------------------------------------

5.Write a function where month and year are taken as arguments which returns the output with all the dates of saturdays occuring the month

--------------------------------------------------------------------------------------------------------------------

import calendar

def satCounter(year ,month):

    days=[0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

    sat=[]

    d=days[month]

    if month==2 and calendar.isleap(year):

        d=29

    for day in range(1,d+1):

        sat.append(calendar.weekday(year,month,day))

    # print(sat)

    return f"number of saturdays are {sat.count(6)}"

year=int(input("Enter the year"))

month=int(input("Enter month 'in numbers':"))

print(satCounter(year,month))

----------------------------------------------------------------------------------------------------------------------

6.

#  Create a dictionary where the key is an even number from the given list and the value

# will be the occurrence of that element in the list. input= [1,2,3,2,4,2,4,7,8,4,5,8,6,9,2]

x={}

input= [1,2,3,2,4,2,4,7,8,4,5,8,6,9,2]

input=sorted(input)

for i in input:

    if i in x:

        continue

    else:

        if i%2==0:

            x.update({i:0})

print(x)

for key in x:

    x[key]=input.count(key)

# withot built in...........................................................

# for key in x:

#     count=0

#     for i in range(len(input)):

#         if input[i]==key:

#             count+=1

#         if input[i]==key+1:

#             break

#     x[key]=count

print(x)

7.

# 9. Write a function swap\_element that contains two args which will be the position of

# elements present in the list. The function must swap the elements present in those

# positions.

# Input: [1,2,3,4,5,6,7,8] function: swap\_element(arg1, arg2)

def swap\_element(x,y):

    input[x],input[y]=input[y],input[x]

input=[1,2,3,4,5,6,7,8]

swap\_element(2,4)

print(input)

1. match = ‘version’, input=’Upgraded\_image\_version\_8.0.4.3’ if match in input: print(‘YES’) else: print(‘NO’)

Ans=”YES”

**9.**

#  Rewrite the program to get proper output

# Match = 'version'

# input=8

# print(Match+input)

match='version'

input=8

print(match+str(input))

**10.**

# Write a program to multiply two given number without using “\*” operation and any in built

# function

def multiply(x,y):

    mi=min(x,y)

    ma=max(x,y)

    total=0

    for i in range(mi):

        total+=ma

    print(total)

multiply(10,2)

multiply(10,0)

**11.**

#  Write a program to find the count of alphabet alone in the given alphanumeric string for

# Ex1: input=’abb24ccc8ddbbca1’ output=’a1b224c38d2b2c1a11’

# Ex2: input = ‘abc23’ output=’a1b1c123’

input='abb24ccc8ddbbca1'

# input='abc23'

op=''

x=''

for i in range(len(input)):

    if not input[i].isalpha():

        op=op+input[i]

    else:

        x=x+input[i]

        if input[i] != input[i+1]:

            op=op+x[0]+str(len(x))

            x=''

print(op)

**12.**Write a python program for sort the given below list based last character of each word names\_list = ['Prabhu', Rahul', 'Arunesh, 'Sonali', 'Rakshit']

# Write a python program for sort the given below list

# based last character of each word names\_list = ['Prabhu', Rahul', 'Arunesh, 'Sonali', 'Rakshit']

l=['Prabhu', 'Rahul', 'Arunesh', 'Sonali', 'Rakshit']

d={}

for i in range(len(l)):

    d.update({l[i][-1]:l[i]})  #This makes d={'u': 'Prabhu', 'l': 'Rahul', 'h': 'Arunesh', 'i': 'Sonali', 't': 'Rakshit'}

    l[i]=l[i][-1] # This makes l=['u', 'l', 'h', 'i', 't']

l.sort()

for i in range(len(l)):

    l[i]=d[l[i]]

print(l)

**13.**Find the highest sum of the string by removing the duplicates for each iteration input=’1211’

**14.**Write a python script to copy files from a directory D1 based on timestamp(current\_date) to another directory D2 and delete the source directory D1. Whenever the script is called this program must run.

**15.**Write a program to send a mail notification to customers regarding the arrival of goods on a daily basis. The admin email has a separate domain email address owned by your company.Do not forget to add cc candidates in customer’s mail

1. You are given a string S. Your task is to find the indices of the start and end of string k in S The first line contains the string S.The second line contains the string k. Print the tuple in this format: (start \_index, end \_index). If no match is found, print (-1, -1). Sample Input Sample Output aaadaa aa (0, 1) (1, 2) (4, 5)

# 26. You are given a string S. Your task is to find the indices of the start and end of string k in

# S The first line contains the string S.The second line contains the string k.

# Print the tuple in this format: (start \_index, end \_index). If no match is found, print (-1,

# -1).

# Sample Input Sample Output

# aaadaa

# aa

# (0, 1)

# (1, 2)

# (4, 5)

s='aaadaa'

k='aa'

def substring(s,k):

    size=len(k)

    l=[]

    for i in range(len(s)-size+1): #length-size+1 because we are slicing from given index to given+size index

        if k==s[i:i+size]:

            l.append((i,i+size-1))

            # print(s[i:i+size])

    return l

print(substring(s,k))

**17.**

# 27. Write a Python class to check the validity of a string of parentheses, '(', ')', '{', '}', '[' and '].

# These brackets must be closed in the correct order, for example "()" and "()[]{}" are valid

# but "[)", "({[)]" and "{{{" are invalid

input=["()", "()[]{}", "[)", "({[)]" ,"{{{" ]

paren={')':'(','}':'{',']':'['}

def parentheses(x):

    l=[]

    for i in x:

        if i not in paren:

            l.append(i)

        else:

            if l and paren[i]==l[-1]:

                l.pop()

            else:

                return False

    # print(l)

    if not l :

        return True

    else:

        return False

# Testing each input.....................

for i in input:

    print(i,parentheses(i))

    print()

**18.**

# Write a Python program to remove the parenthesis area in a string using Regular

# Expression

# Sample data : ["example (.com)", "MSys", "github (.com)", "keka (.com)"]

# Expected Output:

# Example

# MSys

# github

# keka

input= ["example (.com)", "MSys", "github (.com)", "keka (.com)"]

def func(s):

    for i in s:

        if '(' not in s:

            print(s)

            break

        else:

            index=s.index('(')

            print(s[:index])

            break

for i in input:

    func(i)

**19.**

Write a regular expression to find the html tags that are more than 4 letters. Note: Html tags can be found inside <> characters and closing html tags can be found in the same format after / character. </> i.e.: <param> </param>