-TASK:

Program to print the prime numbers between the range 25 to 30

In [1]:

```
a = []
x = int(input("enter the number: "))
y = int(input("enter the number: "))
for i in range(x,y+1):
    if i>1:
        for j in range(2,i):
            if i%j==0:
                #print("not a prime no",i)
                break
        else:
            #print("it is a prime no",i)
            a.append(i)
            #print(a)
    else:
        print("neither prime nor composite",i)
print(a)
```

```
enter the number: 25
enter the number: 50
[29, 31, 37, 41, 43, 47]
```

-TASK:

Roy wants to change his profile picture on Facebook. Now Facebook has some restriction over the dimension of picture that we can upload. Minimum dimension of the picture can be L x L, where L is the length of the side of square.

Now Roy has N photos of various dimensions. Dimension of a photo is denoted as W x H where W - width of the photo and H - Height of the photo

When any photo is uploaded following events may occur:

If any of the width or height is less than L, user is prompted to upload another one. Print "UPLOAD ANOTHER" in this case. If width and height, both are large enough and if the photo is already square then it is accepted. Print "ACCEPTED" in this case. else user is prompted to crop it. Print "CROP IT" in this case. (quotes are only for clarification)

Given L, N, W and H as input, print appropriate text as output.

input:

First line contains L. Second line contains N, number of photos. Following N lines each contains two space separated integers W and H. Output:

Print appropriate text for each photo in a new line.

In [ ]:

```
L = int(input("enter the length of dp: "))
N = int(input("enter the no. of photos to upload: "))
W = int(input("enter the width of dp: "))
H = int(input("enter the height of dp: "))
x = 0
while x<N:
    #x+=1
    if W < L and H > L:
        print("upload another case")
    elif W < L and H < L:
        print("size is small")
    elif W > L and H < L:
        print("not fits well")
    elif W > L and H > L:
        print("crop it")
    elif W == L and H == L:
        print("accepted")
        print("UPLOADING")
    x+=1
```

-TASK:

print the diag elements from right: Input : [[1,2,3],[4,5,6],[7,8,9]] Output : [3,5,7]

In [13]:

```
import numpy as np
D = np.array([[1,2,3],[4,5,6],[7,8,9]])
print(D)
print(np.diag(D))
def left_diag(D):
    l = []
    m = []
    a = len(D)
    for i in D:
        l.append(i[a-1])
        a = a-1
    print(l)
left_diag(D)
```

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

-TASK:

print this pattern [[1,1,1,1,1] [1,0,0,0,1] [1,0,9,0,1] [1,0,0,0,1] [1,1,1,1,1]]

In [15]:

```
x = []
y = int(input("enter the size of an array: "))
z = 0
while z<y:
    z+=1
    a = eval(input("enter anything: "))
    x.append(a)
print(x)
b = np.array(x)
print(b.reshape(5,5))
```

```
enter the size of an array: 25
enter anything: 1
enter anything: 1
enter anything: 1
enter anything: 1
enter anything: 1
enter anything: 1
enter anything: 0
enter anything: 0
enter anything: 0
enter anything: 1
enter anything: 1
enter anything: 0
enter anything: 9
enter anything: 0
enter anything: 1
enter anything: 1
enter anything: 0
enter anything: 0
enter anything: 0
enter anything: 1
enter anything: 1
enter anything: 1
enter anything: 1
enter anything: 1
enter anything: 1
[1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 9, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,
1]
[[1 1 1 1 1]
 [1 0 0 0 1]
 [1 0 9 0 1]
 [1 0 0 0 1]
 [1 1 1 1 1]]
```

In [16]:

```
x = np.zeros((5,5),dtype=int)
x[0,0:]=1
x[1:,0]=1
x[4,0:]=1
x[1:,4]=1
x[2:3,2]=9
print(x)
```

```
[[1 1 1 1 1]
 [1 0 0 0 1]
 [1 0 9 0 1]
 [1 0 0 0 1]
 [1 1 1 1 1]]
```

-TASK:

given an integer "n" perform the following conditional actions

if n is odd,print weird if n is even and in the inclusive range of 2 to 5,print not weird if n is even and in the inclusive range of 6 to 20, print weird if n is even and greater than 20,print not weird

In [1]:

```
n = int(input("enter the number: "))
if n % 2 != 0:
    print("weird")
elif n % 2 == 0:
    if 0<n<6:
        print("not weird")
    elif 4<n<22:
        print("weird")
    elif n>20:
        print("not weird")
    else:
        print("error")
```

```
enter the number: 5
weird
```

-TASK:

Given an list of integers nums and add an integer target return indicates of the two numbers such that they add up to get the target..

INPUT 1: nums=[2,7,11,15] , target=9 OUTPUT 1: [0,1] (Because nums[0]+nums[1]==9, we return [0,1])

INPUT 2: nums=[3,2,4] , target=6 OUTPUT: [1,2]

In [2]:

```
num = []
l = int(input("enter the length of the list: "))
k = 0
z = 0
target = 9
while k < l:
    k+=1
    m = int(input("enter anything: "))
    num.append(m)
print(num)
for i in range(0,len(num)):
    if num[i]+num[i+1]==target:
        z+=1
        print("the numbers whose sum is 9 are: ",num[i],num[i+1])
        print("the index of those nos are: ",i,i+1)
```

```
enter the length of the list: 4
enter anything: 2
enter anything: 7
enter anything: 11
enter anything: 15
[2, 7, 11, 15]
2 7
0 1

---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
Input In [2], in <cell line: 11>()
     10 print(num)
     11 for i in range(0,len(num)):
---> 12         if num[i]+num[i+1]==target:
     13                 z+=1
     14                 print(num[i],num[i+1])

IndexError: list index out of range
```

-TASK:

current no 0 previous no 0 current no 1 previous no 0 current no 2 previous no 1 current no 3 previous no 2 current no 4 previous no 3 current no 5 previous no 4 current no 6 previous no 5 current no 7 previous no 6 current no 8 previous no 7 current no 9 previous no 8

In [7]:

```
x = 0
for i in range(10):
    if i == 0:
        print("current no",i,"previous no",i)
    elif i >= 0:
        print("current no",i,"previous no",i-1)
```

```
current no 0 previous no 0
current no 1 previous no 0
current no 2 previous no 1
current no 3 previous no 2
current no 4 previous no 3
current no 5 previous no 4
current no 6 previous no 5
current no 7 previous no 6
current no 8 previous no 7
current no 9 previous no 8
```

-TASK:

run a program to capitalize the first and last character of each word in a string

input = hello world output = HellO WorlD

In [8]:

```
x = "muskaan verma"
print(x[0].upper()+x[1:6]+x[6].upper(),x[8].upper()+x[9:12]+x[-1].upper())
```

```
MuskaaN VermA
```

-TASK:

x = "ishita" remove "i" from this string output - "shta"

In [3]:

```
x = "ishita"
for i in x:
    if i == "i":
        continue
    print(i,end="")
```

```
shta
```

In [4]:

```
x = "ishita"
y = x.replace("i","")
print(y)
```

```
shta
```

-TASK:

lets learn about list comprehensions you are given three integers x,y,and z representing the dimensions of cuboid along with an integer "n" print a list of all possible coordinates given by (i,j,k) on a 3D grid where the sum of i+j+k is not equal to n here, 0<=i<=x, i can only take values from 0 till x 0<=j<=y, j can only take values from 0 till y 0<=k<=z k can only take values from 0 till z

input: x = 1 y = 1 z = 2 n = 3 all permutations of [i,j,k] are: [[0,0,0],[0,0,1],[0,0,2],[0,1,0],[0,1,1],[0,1,2],[1,0,0],[1,0,1],[1,0,2]]

but here we have to print the list of permutations of [i,j,k] which is not equal to n so we will take only these arrays:

[[0,0,0],[0,0,1],[0,0,2],[0,1,0],[0,1,1],[1,0,0],[1,0,1] etc]

In [12]:

```
a = []
x = int(input("enter the no1: "))
y = int(input("enter the no2: "))
z = int(input("enter the no3: "))
n = int(input("enter an integer: "))
for i in range(0,x+1):
    for j in range(0,y+1):
        for k in range(0,z+1):
            if i+j+k!=n:
                b = [i,j,k]
                a.append(b)
print(a)
```

```
enter the no1: 1
enter the no2: 1
enter the no3: 2
enter an integer: 3
[[0, 0, 0], [0, 0, 1], [0, 0, 2], [0, 1, 0], [0, 1, 1], [1, 0, 0], [1, 0,
1], [1, 1, 0], [1, 1, 2]]
```

-TASK:

given the participants score sheets for your university sports day, you are required to find the runner up score you are given n scores store them in a list and find the score of the runner up

output - [2,3,6,6,5] runner up score - 5

In [5]:

```
a = []
x = int(input("enter the number of student: "))
for i in range(x):
    n = int(input("enter the scores: "))
    a.append(n)
b = sorted(a,reverse=True)
print(b)
for i in b:
    count = 0
    for j in b:
        if i == j:
            count+=1
            if count>1:
                b.remove(j)
print(b)
print("runnerupscore: ",b[1])
```

```
enter the number of student: 5
enter the scores: 1
enter the scores: 2
enter the scores: 3
enter the scores: 4
enter the scores: 5
[5, 4, 3, 2, 1]
[5, 4, 3, 2, 1]
runnerupscore:  4
```

-TASK:

print the even places words of the word "pynative"

In [7]:

```
x = "pynative"
for i in range(len(x)):
    if i%2==0:
        print(x[i],end="")
```

```
pntv
```

In [8]:

```
x = "pynative"
for i in range(0,len(x),2):
    print(x[i],end="")
```

```
pntv
```

-TASK:

write a func. to return True if the first and last number of the given list is same if the nos are different then return false

> given: number_x = [10,20,30,40,10] true number_y = [75,65,55,32,45] false

In [12]:

```
x = []
y = int(input("enter the size of the list: "))
z = 0
while z < y:
    z+=1
    a = eval(input("enter anything: "))
    x.append(a)
print(x)
if x[0]==x[-1]:
    print("true")
else:
    print("false")
```

```
enter the size of the list: 5
enter anything: 12
enter anything: 30
enter anything: 55
enter anything: 66
enter anything: 77
[12, 30, 55, 66, 77]
false
```

-TASK:

print a dict by user input then {2:56,1:2,5:12,4:24,6:18,3:323} keys and values are sorted in alphabetical order by keys it should be arranged in thus way (1:2),(2:56),(3:323),(4:24) etc..

In [13]:

```
x = {}
ans = []
y = int(input("enter the size of dict: "))
z = 0
while z < y:
    z+=1
    a = eval(input("enter the keys: "))
    b = eval(input("enter the values: "))
    x.update({a:b})
    ans.append([a,b])
print("dict as follows",x)
print(ans)
s = sorted(x.keys())
print("the keys of the dict are: ",s)
result = sorted(ans)
print(result)
m = dict(result)
print(m)
```

```
enter the size of dict: 6
enter the keys: 2
enter the values: 56
enter the keys: 1
enter the values: 2
enter the keys: 5
enter the values: 12
enter the keys: 4
enter the values: 24
enter the keys: 6
enter the values: 18
enter the keys: 3
enter the values: 323
dict as follows {2: 56, 1: 2, 5: 12, 4: 24, 6: 18, 3: 323}
[[2, 56], [1, 2], [5, 12], [4, 24], [6, 18], [3, 323]]
the keys of the dict are:  [1, 2, 3, 4, 5, 6]
[[1, 2], [2, 56], [3, 323], [4, 24], [5, 12], [6, 18]]
{1: 2, 2: 56, 3: 323, 4: 24, 5: 12, 6: 18}
```

-TASK:

Create a new list from a two list using the following condition

Given a two list of numbers, write a program to create a new list such that the new list should contain odd numbers from the first list and even numbers from the second list.

Given:

list1 = [10, 20, 25, 30, 35] list2 = [40, 45, 60, 75, 90] Expected Output:

result list: [25, 35, 40, 60, 90]

In [14]:

```
list1 = []
list2 = []
list3 = []
x = int(input("enter the length of list1: "))
y = int(input("enter the length of list2: "))
z1 = 0
z2 = 0
while z1<x:
    z1+=1
    a = eval(input("enter anything: "))
    list1.append(a)
print(list1)
while z2<y:
    z2+=1
    b = eval(input("enter items: "))
    list2.append(b)
print(list2)
for i in range(len(list1)):
    if list1[i]%2!=0:
        print("odd nos are",list1[i],end=" ")
        list3.append(list1[i])
        print()
for j in range(len(list2)):
    if list2[j]%2==0:
        print("even nos are",list2[j],end=" ")
        list3.append(list2[j])
        print()
print(list3)
```

```
enter the length of list1: 5
enter the length of list2: 5
enter anything: 10
enter anything: 20
enter anything: 25
enter anything: 20
enter anything: 35
[10, 20, 25, 20, 35]
enter items: 40
enter items: 45
enter items: 60
enter items: 75
enter items: 90
[40, 45, 60, 75, 90]
odd nos are 25
odd nos are 35
even nos are 40
even nos are 60
even nos are 90
[25, 35, 40, 60, 90]
```

-TASK:

create a program to cheach a no. is prime or not

In [16]:

```
x = int(input("enter a number: "))
y = 2
if y > x:
    print("number is not a prime no.")
elif x == y:
    print("number is a prime number")
else:
    while y < x:
        if x % y == 0:
            print("composite number")
            break
        y += 1
    else:
        print("it is prime no.")
```

```
enter a number: 4
composite number
```

-TASK:

input = ["apple","orange","lichi","lemon","plum"] newlist = [] output newlist = ["lichi","lemon"]

In [2]:

```
x = ["apple","orange","lichi","lemon","plum"]
y=x[2:4]
print(y)
```

```
['lichi', 'lemon']
```

-TASK:

Find the sum of the series upto n terms Write a program to calculate the sum of series up to n term. For example, if n =5 the series will become 2 + 22 + 222 + 2222 + 22222 = 24690 #2[1+11+111+1111+11111]

In [25]:

```
x = int(input("enter the n: "))
y = 0
summ=0
a = int(input("enter the number: "))
for i in range(x):
    y=(y*10+1)
    print(y*a,end=" ")
    print()
    summ+=(y*a)
print("the sum of the above series is: ",summ)
```

```
enter the n: 5
enter the number: 2
2
22
222
2222
22222
the sum of the above series is:  24690
```

-TASK:

Write a program which can compute the factorial of a given numbers. The results should be printed in a comma-separated sequence on a single line. Suppose the input is supplied to the program: 8 Then, the output should be: 40320

In [26]:

```
x = int(input("enter the number: "))
y = 1
while x>0:
    y = x*y
    x-=1
print(y)
```

```
enter the number: 4
24
```

-TASK:

With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that is an integral number between 1 and n (both included) and then the program should print the dictionary. Suppose the input is supplied to the program: 8 Then, the output should be: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

In [27]:

```
a = {}
n = int(input("enter the length of the dict: "))
b = 0
while b<n:
    b+=1
    c = int(input("enter the keys: "))
    d = c*c
    a.update({c:d})
print(a)
```

```
enter the length of the dict: 8
enter the keys: 1
enter the keys: 2
enter the keys: 3
enter the keys: 4
enter the keys: 5
enter the keys: 6
enter the keys: 7
enter the keys: 8
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

-TASK:

Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number. Suppose the input is supplied to the program: 34, 67, 55, 33, 12, 98 Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34', '67', '55', '33', '12', '98')

In [28]:

```
x = []
y = int(input("enter the length of list: "))
z = 0
while z<y:
    z+=1
    a = eval(input("enter the input: "))
    x.append(a)
print(x)
b = tuple(x)
print(b)
```

```
enter the length of list: 6
enter the input: 34
enter the input: 67
enter the input: 55
enter the input: 33
enter the input: 12
enter the input: 98
[34, 67, 55, 33, 12, 98]
(34, 67, 55, 33, 12, 98)
```

-TASK:

Write a program that calculates and prints the value according to the given formula: Q = Square root of [(2 * C * D)/H] Following are the fixed values of C and H: C is 50. H is 30. D is the variable whose values should be input to your program in a comma-separated sequence. Let us assume the following comma separated input sequence is given to the program: 100,150,180 The output of the program should be: 18, 22, 24 Hints: If the output received is in decimal form, it should be rounded off to its nearest value (for example, if the output received is 26.0, it should be printed as 26) In case of input data being supplied to the question, it should be assumed to be a console input.

In [33]:

```
from math import sqrt

c = 50
h = 30
a = []
n = []
b = int(input("enter the length of the list: "))
m = 0
while m<b:
    m+=1
    d = int(input("enter the number: "))
    a.append(d)
print(a)
for i in range(len(a)):
    e = sqrt((2*c*a[i])/h)
    print(e)
    f = round(e)
    print(f)
    n.append(f)
print(n)
```

```
enter the length of the list: 3
enter the number: 100
enter the number: 150
enter the number: 180
[100, 150, 180]
18.257418583505537
18
22.360679774997898
22
24.49489742783178
24
[18, 22, 24]
```

-TASK:

Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically. Suppose the input is supplied to the program: without, hello, bag, world Then, the output should be: bag, hello, without, world

In [2]:

```
z = []
x = int(input("enter how many strings u want to enter: "))
y = 0
while y<x:
    y+=1
    a = str(input("enter the input: "))
    z.append(a)
print(z)
b = sorted(z)
print(b)
```

```
enter how many strings u want to enter: 4
enter the input: without
enter the input: hello
enter the input: bag
enter the input: world
['without', 'hello', 'bag', 'world']
['bag', 'hello', 'without', 'world']
```

-TASK:

Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized.
Suppose the input is supplied to the program: Hello world Practice makes perfect Then, the output should be: HELLO WORLD PRACTICE MAKES PERFECT

In [3]:

```
x = str(input("enter the sentence: "))
print(x.upper())
```

```
enter the sentence: hello world
HELLO WORLD
```

-TASK:

Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. Suppose the input is supplied to the program: hello world and practice makes perfect and hello world again Then, the output should be: again and hello makes perfect practice world

In [2]:

```
z = []
a = int(input("enter the length of list: "))
b = 0
while b<a:
    b+=1
    y = str(input("enter the input: "))
    z.append(y)
print(z)
for i in z:
    count= 0
    for j in z:
        if i == j:
            count+=1
            if count>1:
                z.remove(j)
print(z)
m = sorted(z)
print(m)
str1 = " "
for n in m:
    str1+=n
    print(n,end=" ")
```

```
enter the length of list: 10
enter the input: hello
enter the input: world
enter the input: and
enter the input: practice
enter the input: makes
enter the input: perfect
enter the input: and
enter the input: hello
enter the input: world
enter the input: again
['hello', 'world', 'and', 'practice', 'makes', 'perfect', 'and ', 'hello',
'world', 'again']
['and', 'practice', 'makes', 'perfect', 'and ', 'hello', 'world', 'again']
['again', 'and', 'and ', 'hello', 'makes', 'perfect', 'practice', 'world']
again and and  hello makes perfect practice world
```

-TASK:

Write a program which takes a input as a string and print * at all the even indexes Suppose the input is supplied to the program: deepanshu Then the output should be: *epnh**

In [9]:

```
x = (input("enter the name: "))
y = len(x)
for i in range(0,y):
    if i%2==0:
        z = x.replace(x[i],"*")
        print(z)
```

```
enter the name: muskaan
*uskaan
mu*kaan
musk**n
muskaa*
```

-TASK:

Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number. The numbers obtained should be printed in a comma-separated sequence on a single line.

In [11]:

```
a = []
for i in range(1000, 3001):
    s = str(i)
    if (int(s[0])%2==0) and (int(s[1])%2==0) and (int(s[2])%2==0) and (int(s[3])%2==0):
        a.append(s)
print(a)
```

```
['2000', '2002', '2004', '2006', '2008', '2020', '2022', '2024', '2026',
'2028', '2040', '2042', '2044', '2046', '2048', '2060', '2062', '2064', '2
066', '2068', '2080', '2082', '2084', '2086', '2088', '2200', '2202', '220
4', '2206', '2208', '2220', '2222', '2224', '2226', '2228', '2240', '224
2', '2244', '2246', '2248', '2260', '2262', '2264', '2266', '2268', '228
0', '2282', '2284', '2286', '2288', '2400', '2402', '2404', '2406', '240
8', '2420', '2422', '2424', '2426', '2428', '2440', '2442', '2444', '244
6', '2448', '2460', '2462', '2464', '2466', '2468', '2480', '2482', '248
4', '2486', '2488', '2600', '2602', '2604', '2606', '2608', '2620', '262
2', '2624', '2626', '2628', '2640', '2642', '2644', '2646', '2648', '266
0', '2662', '2664', '2666', '2668', '2680', '2682', '2684', '2686', '268
8', '2800', '2802', '2804', '2806', '2808', '2820', '2822', '2824', '282
6', '2828', '2840', '2842', '2844', '2846', '2848', '2860', '2862', '286
4', '2866', '2868', '2880', '2882', '2884', '2886', '2888']
```

-TASK:

Write a program that accepts a sentence and calculate the number of letters and digits. Suppose the input is supplied to the program: hello world! 123 Then, the output should be: LETTERS 10 DIGITS 3

In [18]:

```
j = input("enter the str: ")
count_num = 0
count_letter = 0
for i in j:
    if i.isdigit():
        count_num+=1
    if i.isalpha():
        count_letter+=1
print("digits",count_num)
print("letters",count_letter)
```

```
enter the str: hello world123
digits 3
letters 10
```

-TASK:

Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters. Suppose the input is supplied to the program: Hello world! Then, the output should be: UPPER CASE 1 LOWER CASE 9

In [19]:

```
j = input("enter the str: ")
count_lower = 0
count_upper = 0
for i in j:
    if i.islower():
        count_lower+=1
    if i.isupper():
        count_upper+=1
print("letters in lower case ",count_lower)
print("letters in upper case",count_upper)
```

```
enter the str: hello worlD
letters in lower case  9
letters in upper case 1
```

-TASK:

Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers. Suppose the input is supplied to the program: 1,2,3,4,5,6,7,8,9 Then, the output should be: 1, 3, 5, 7, 9 1,2,9,4,25,6,49,64,81

In [49]:

```
x = []
k = []
m = []
y = int(input("enter the length of the list: "))
z = 0
while z<y:
    z+=1
    a = int(input("enter the input: "))
    x.append(a)
print(x)
for i in range(len(x)):
    #print(i)
    if x[i]%2==0:
        m.append(x[i])
    else:
        k.append(x[i])
        x[i]=x[i]*x[i]
        m.append(x[i])

print(m)
print(k)
```

```
enter the length of the list: 9
enter the input: 1
enter the input: 2
enter the input: 3
enter the input: 4
enter the input: 5
enter the input: 6
enter the input: 7
enter the input: 8
enter the input: 9
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 9, 4, 25, 6, 49, 8, 81]
[1, 3, 5, 7, 9]
```

-TASK:

Write a Python program that accepts a list of integers and calculates the length and the fifth element. Return true if the length of the list is 8 and the fifth element occurs thrice in the said list. Input: [19, 19, 15, 5, 5, 5, 1, 2] Output: True Input: [19, 15, 5, 7, 5, 5, 2] Output: False

In [10]:

```
x = []
y = int(input("enter the length of the list: "))
z = 0
while z<y:
    z+=1
    a = int(input("enter the number: "))
    x.append(a)
print(x)
for i in x:
    f = x[4]
    if len(x)==8 and x.count(f)==3:
        print("true")
    else:
        print("false")
```

```
enter the length of the list: 8
enter the number: 1
enter the number: 2
enter the number: 3
enter the number: 4
enter the number: 5
enter the number: 6
enter the number: 7
enter the number: 8
[1, 2, 3, 4, 5, 6, 7, 8]
false
false
false
false
false
false
false
false
```

-TASK:

Write a Python program to find the length of a given list of non-empty strings.
Input: ['cat', 'car', 'fear', 'center'] Output: [3, 3, 4, 6]

Input: ['cat', 'dog', 'shatter', 'donut', 'at', 'todo', ''] Output: [3, 3, 7, 5, 2, 4, 0]

In [14]:

```
a1 = []
x = []
y = int(input("enter the length of the list: "))
z = 0
while z<y:
    z+=1
    a = input("enter the word: ")
    x.append(a)
print(x)
str1 = " "

for i in x:
    str1+=i
    print(i,end=" ")
    a1.append(len(i))
print(a1)
```

```
enter the length of the list: 7
enter the word: cat
enter the word: dog
enter the word: shatter
enter the word: donut
enter the word: at
enter the word: todo
enter the word: 
['cat', 'dog', 'shatter', 'donut', 'at ', 'todo', '']
cat dog shatter donut at  todo  [3, 3, 7, 5, 3, 4, 0]
```

In [15]:

```
n = int(input("enter the length: "))
l = []
x = []
for i in range(n):
    words = input("enter the words: ")
    l.append(words)
print(l)
for i in l:
    x.append(len(i))
print(x)
```

```
enter the length: 7
enter the words: cat
enter the words: dog
enter the words: shatter
enter the words: donut
enter the words: at
enter the words: todo
enter the words: 
['cat', 'dog', 'shatter', 'donut', 'at ', 'todo', '']
[3, 3, 7, 5, 3, 4, 0]
```

-TASK:

Write a Python program to find strings in a given list starting with a given prefix. Go to the editor Input: [(ca,('cat', 'car', 'fear', 'center'))] Output: ['cat', 'car'] Input: [(do,('cat', 'dog', 'shatter', 'donut', 'at', 'todo'))]

> Output: ['dog', 'donut']

In [31]:

```
a1 = []
x = []
y = int(input("enter the length of the list: "))
z = 0
while z<y:
    z+=1
    a = input("enter the str: ")
    x.append(a)
print(x)
s = input("enter the input: ")
for i in x:
    if i[0] + i[1] == s:
        a1.append(i)
print(a1)
```

```
enter the length of the list: 4
enter the str: car
enter the str: cat
enter the str: cash
enter the str: center
['car', 'cat', 'cash', 'center']
enter the input: ca
['car', 'cat', 'cash']
```

In [33]:

```
a1 = []
x = []
y = int(input("enter the length of the list: "))
z = 0
while z<y:
    z+=1
    a = input("enter the str: ")
    x.append(a)
print(x)
str1 = " "
for j in x:
    str1+=j
    #print(j)
    if (j[0]+j[1]).startswith("ca"):
        a1.append(j)
print(a1)
```

```
enter the length of the list: 4
enter the str: cash
enter the str: cat
enter the str: car
enter the str: center
['cash', 'cat', 'car', 'center']
['cash', 'cat', 'car']
```

-TASK:

In [34]:

```
x = [-1,2,0,-9,-4,3,7,5,6,2,-2,1]
a = []
b1 = []
c = []
y = sorted(x)
print("sorted list in asc order:", y)
for i in x:
    if i>0:
        a.append(i)
        a.sort()
    else:
        b1.append(i)
print("all the negative nos are: ",b1)
#print("all the positive nos are: ",a)
for m in a:
    count = 0
    for n in a:
        if m == n:
            count+=1
            if count>1:
                a.remove(n)
c.append(a)
print("all the +ve nos without duplicates",c)
b = a[0]
for f in a :
      if f > b :
        b = f
m = a[0]
for f in a:
    if f < m:
        m = f
missing = b+1
list1=[]
for _ in a :
    b = b -1
    if b not in a :
        list1.append(b)
print(list1)
```

```
sorted list in asc order: [-9, -4, -2, -1, 0, 1, 2, 2, 3, 5, 6, 7]
all the negative nos are:  [-1, 0, -9, -4, -2]
all the +ve nos without duplicates [[1, 2, 3, 5, 6, 7]]
[4]
```

In [35]:

```
li = []
l =  [-1,2,0,-9,-4,3,7,5,6,2,-2,1]
for i in l:
    if i>0:
        li.append(i)
        li.sort()
for i in range(min(li),max(li)):
    if i not in li:
        print(i)
```

```
4
```

In [37]:

```
x = []
y = int(input("enter the length of the list: "))
z = 0
while z < y:
    z+=1
    a = eval(input("enter anything: "))
    x.append(a)
print(x)
s = sorted(x)
print(s)
m = []
for i in range(s[0],s[-1]+1):
    m.append(i)
print(m)
u = set(s)
v = set(m)
p = u.symmetric_difference(v)
print("missing values",p)
q = list(p)
print("the missing least positive no:",q[0])
```

```
enter the length of the list: 12
enter anything: -1
enter anything: 2
enter anything: 0
enter anything: -9
enter anything: -4
enter anything: 3
enter anything: 7
enter anything: 5
enter anything: 6
enter anything: 2
enter anything: -2
enter anything: 1
[-1, 2, 0, -9, -4, 3, 7, 5, 6, 2, -2, 1]
[-9, -4, -2, -1, 0, 1, 2, 2, 3, 5, 6, 7]
[-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7]
missing values {4, -8, -7, -6, -5, -3}
the missing least positive no: 4
```

-TASK:

Write a Python program to determine the direction ('increasing' or 'decreasing') of monotonic sequence numbers. Go to the editor Input: [1, 2, 3, 4, 5, 6] Output: Increasing. Input: [6, 5, 4, 3, 2, 1] Output: Decreasing.

In [38]:

```python
x = []
y = int(input("enter the length of the list: "))
for i in range(y):
    a = int(input("enter the values: "))
    x.append(a)
print(x)
if x[0]==min(x):
    print("increasing order")
else:
    print("decreasing order")
```

```
enter the length of the list: 6
enter the values: 6
enter the values: 5
enter the values: 4
enter the values: 3
enter the values: 2
enter the values: 1
[6, 5, 4, 3, 2, 1]
decreasing order
```

-TASK:

Write a Python program to determine which triples sum to zero from a given list of lists. Input: [[1343532, -2920635, 332], [-27, 18, 9], [4, 0, -4], [2, 2, 2], [-20, 16, 4]] Output: [False, True, True, False, True] Input: [[1, 2, -3], [-4, 0, 4], [0, 1, -5], [1, 1, 1], [-2, 4, -1]] Output: [True, True, False, False, False]

In [41]:

```
x = []
y = int(input("enter the length of the list: "))
z = 0
while z<y:
    z+=1
    a = []
    b = int(input("enter the length of the list2: "))
    c = 0
    while c<b:
        c+=1
        d = int(input("enter the numbers: "))
        a.append(d)
    x.append(a)
print(x)
s = []
for i in x:
    if i[0]+i[1]+i[2]==0:
        result="true"
    else:
        result="false"
    s.append(result)
print(s)
```

```
enter the length of the list: 5
enter the length of the list2: 3
enter the numbers: 1
enter the numbers: 2
enter the numbers: -3
enter the length of the list2: 3
enter the numbers: -4
enter the numbers: 0
enter the numbers: 4
enter the length of the list2: 3
enter the numbers: 0
enter the numbers: 1
enter the numbers: -5
enter the length of the list2: 3
enter the numbers: 1
enter the numbers: 1
enter the numbers: 1
enter the length of the list2: 3
enter the numbers: -2
enter the numbers: 4
enter the numbers: -1
[[1, 2, -3], [-4, 0, 4], [0, 1, -5], [1, 1, 1], [-2, 4, -1]]
['true', 'true', 'false', 'false', 'false']
```

-TASK:

Write a Python program to sort numbers based on strings. Input: six one four one two three Output: one two three four six Input: six one four three two nine eight Output: one two three four six eight nine Input: nine eight seven six five four three two one Output: one two three four five six seven eight nine

In [ ]:

```
n = int(input("enter the length: "))
l1 = ["one","two","three","four","five","six","seven","eight","nine"]
l2 = [1,2,3,4,5,6,7,8,9]
z = zip(l1,l2)
l = list(z)
nums = []
#print(l)
for i in range(n):
    numbers = str(input("enter the strings: "))
    nums.append(numbers)
print(nums)
final = []
final1 = []
for j in nums:
    for a,b in l:
        if a == j:
            final.append(b)
#print(final)
final.sort()
for k in final:
    for x,y in l:
        if y == k:
            final1.append(x)
print(final1)
```

-TASK:

Write a Python program to find all words in a given string with n consonants. Go to the editor Input: this is our time Output: Number of consonants: 3 Words in the said string with 3 consonants: ['this'] Number of consonants: 2 Words in the said string with 2 consonants: ['time'] Number of consonants: 1 Words in the said string with 1 consonants: ['is', 'our']

In [ ]:

```
x = []
vowels = ["a","e","i","o","u"]
y = int(input("enter the length of the list: "))
z = 0
while z<y:
    z+=1
    a = str(input("enter the words: "))
    x.append(a)
print(x)
for i in range(len(x)):
    for j in vowels:
        x[i] = x[i].replace(j,"")
print(x)
str1=" "
for m in x:
    str1+=m
    print(m,end=" ")
    print(len(m))
```

-TASK:

Write a Python program to find even-length words from a given list of words and sort them by length.
Go to the editor
Original list of words: ['Red', 'Black', 'White', 'Green', 'Pink', 'Orange']

Find the even-length words and sort them by length in the said list of words: ['Pink', 'Orange']

Original list of words: ['The', 'worm', 'ate', 'a', 'bird', 'imagine', 'that', '!', 'Absurd', '!!']

Find the even-length words and sort them by length in the said list of words: ['!!', 'bird', 'that', 'worm', 'Absurd']

In [45]:

```
m = []
x = []
y = int(input("enter the length of the list: "))
z = 0
while z<y:
    z+=1
    a = str(input("enter the strings: "))
    x.append(a)
print(x)
str1= " "
for i in x:
    str1+=i
    b = len(i)
    #print(i," ",b)
    if b%2==0:
        print(i,end=" ")
        m.append(i)
print(m)
n = sorted(m,key=len)
print(n)
```

```
enter the length of the list: 6
enter the strings: red
enter the strings: black
enter the strings: white
enter the strings: greenb
enter the strings: pink
enter the strings: orange
['red', 'black', 'white', 'greenb', 'pink', 'orange']
greenb pink orange ['greenb', 'pink', 'orange']
['pink', 'greenb', 'orange']
```

-TASK:

Write a Python program to find the following strange sort of list of numbers: the first element is the smallest, the second is the largest of the remaining, the third is the smallest of the remaining, the fourth is the largest of the remaining, etc. Input: [1, 3, 4, 5, 11] Output: [1, 11, 3, 5, 4] Input: [27, 3, 8, 5, 1, 31] Output: [1, 31, 3, 27, 5, 8] Input: [1, 2, 7, 3, 4, 5, 6] Output: [1, 7, 2, 6, 3, 5, 4]

In [52]:

```
a = []
x = [1, 3, 4, 5, 11]
max_no=0
for i in x:
    if i>max_no:
        max_num = i
print(max_num)
```

```
11
```

-TASK:

print the prime nos from the list input = [1,2,3,4,5] output

In [53]:

```
l = []
n = int(input("enter the length: "))
for i in range(n):
    nums = int(input("enter the numbers: "))
    l.append(nums)
print(l)
l1 = []
count = 0
for j in l:
    a = 2
    if j<a:
        count+=1
        if count == len(l):
            print("no prime numbers found in the list")
    elif j == a:
        l1.append(j)
    elif j>a:
        while a<j:
            if j%a==0:
                break
            a+=1
        else:
            l1.append(j)
print(l1)
```

```
enter the length: 5
enter the numbers: 1
enter the numbers: 3
enter the numbers: 4
enter the numbers: 5
enter the numbers: 11
[1, 3, 4, 5, 11]
[3, 5, 11]
```

-TASK:

given a number n ,for each integer i in range from 1 to n inclusive print one value per line as follows: if i is a multiple of both 3 and 5 print fizzbuzz if i is a multiple of 3 but not 5 print fizz if i is a multiple of 5 but not 3 print buzz if i is not a multiple of 3 or 5 print the value of i

In [2]:

```
def fizzBuzz(n):
    for i in range(1,n):

        if i%3!=0:
            if i%5!=0:
                print(i)
            else:
                print("Buzz")
        elif i%3==0:
            if i%5==0:
                print("FizzBuzz")
            else:
                print("Fizz")
fizzBuzz(16)
```

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
```

-TASK:

write a python program to concatenated two strings with uncommon character Sample Input str1: "12345" str2: "345678"

Sample OUTPUT : "12678"

In [3]:

```
str1 = '12345'
str2 = '345678'
str3 = set(str1)
str4 = set(str2)
print(str3)
print(str4)
x = str3.symmetric_difference(str4)
print(str(x))
#print(''.join(x))
y = sorted(x)
print(y)
print(''.join(y))
```

```
{'1', '3', '2', '4', '5'}
{'7', '3', '4', '5', '6', '8'}
{'1', '2', '7', '6', '8'}
['1', '2', '6', '7', '8']
12678
```

-TASK:

p
p y

p y t

p y t h

p y t h o

p y t h o n

p y t h o

p y t h

p y t

p y

p

In [5]:

```python
x = "python"
for i in range(0,6):
    print()
    for j in range(0,i+1):
        print(x[j],end=" ")
    print()
for i in range(6,1,-1):
    print()
    for j in range(0,i-1):
        print(x[j],end=" ")
    print()
```

```

p 

p y 

p y t 

p y t h 

p y t h o 

p y t h o n 

p y t h o 

p y t h 

p y t 

p y 

p 
```