In [3]:
```
# Print duplicates present in the given list

l = [10,20,30,10,50,20,'yu',6,'yu']
d = []
for i in l:
    if l.count(i)>1 and i not in d:
        d.append(i)
print(d)
```

```
[10, 20, 'yu']
```

In [6]:
```
# Print duplicates present in the given list without using any built in functi
d=[]
for i in range(len(l)):
    for j in range(i+1,len(l)):
        if l[i]==l[j]:
            d.append(l[i])

print(d)
```

```
[10, 20, 'yu']
```

In [10]:
```
for j in range(5):
    for i in range(5):
        print('*',end='')
```

```
*************************
```

In [17]:
```
# SQUARE
for j in range(5): # This loop goes into rows
    for i in range(5): # This loop goes into columns
        print('*',end=' ')#here is space so that there is space bwtween each s
    print()
```

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

In [19]:
```
# Increasing Triangle
for j in range(5): # This loop goes into rows
    for i in range(j+1): # This loop goes into columns
        print('*',end=' ')#here is space so that there is space bwtween each s
    print()
```

```
*
* *
* * *
* * * *
* * * * *
```

In [24]:
```
# Decreasing Triangle
for j in range(5): # This loop goes into rows
    for i in range(j,5): # This loop goes into columns
        print('*',end=' ')#here is space so that there is space bwtween each s
    print()
```

```
* * * * *
* * * *
* * *
* *
*
```

In [32]:
```
# Right Sided Triangle
n = 5
for i in range(n):
    for j in range(i,n):
        print(' ',end=' ')

    for j in range(i+1):
        print('0',end=' ')
    print()
```

```
          0
        0 0
      0 0 0
    0 0 0 0
  0 0 0 0 0
```

In [35]:
```
for i in range(5,1,-1):
    for j in range(5,1,-1):
        print(i,end='')
    print()
```

```
5555
4444
3333
2222
```

In [37]:
```
n = 5
for i in range(n):
    for j in range(i+1):
        print(' ', end=' ')
    for j in range(i,n):
        print('*',end=' ')
    print()
```

```
  * * * * *
    * * * *
      * * *
        * *
          *
```

In [39]:
```
# Hill Pattern
n = 5
for i in range(n):
    for j in range(i,n):
        print(' ',end=' ')

    for j in range(i+1):
        print('*', end=' ')

    for j in range(i):
        print('*',end=' ')

    print()
```

```
          *
        * * *
      * * * * *
    * * * * * * *
  * * * * * * * * *
```

In [41]:
```
# Reverse hill pattern
n = 5
for i in range(n):
    for j in range(i+1):
        print(' ', end=' ')
    for j in range(i,n):
        print('*',end=' ')
    for j in range(i,n-1):
        print('*', end=' ')

    print()
```

```
  * * * * * * * * *
    * * * * * * *
      * * * * *
        * * *
          *
```

In [45]:
```
n = 5
for i in range(n-1):
    for j in range(i,n):
        print(' ',end=' ')

    for j in range(i+1):
        print('*', end=' ')

    for j in range(i):
        print('*',end=' ')

    print()

for i in range(n):
    for j in range(i+1):
        print(' ', end=' ')
    for j in range(i,n):
        print('*',end=' ')
    for j in range(i,n-1):
        print('*', end=' ')

    print()
```

```
          *
        * * *
      * * * * *
    * * * * * * *
  * * * * * * * * *
    * * * * * * *
      * * * * *
        * * *
          *
```

Write program to check value is type list

In [2]:
```
# Write program to check value is type list.
data = {'jay':['male',2000], 'kiran':'male','vijay':['male',1998],'Sanskriti':
count = 0
for i in data:
    if isinstance(data[i],list):
        count+=1
print(count)
```

```
3
```

len(data)

In [51]:
```
count = 0
while i<range(len(data)):
    if type(data[i]) == str:
        count+=1
        i+=1
print(count)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-51-127401d35f35> in <module>
      1 count = 0
----> 2 while i<range(len(data)):
      3     if type(data[i]) == str:
      4         count+=1
      5         i+=1

TypeError: '<' not supported between instances of 'str' and 'range'
```

In [52]:
```
data[1]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-52-c402bf503b75> in <module>
----> 1 data[1]

KeyError: 1
```

In [53]:
```
data.values()
```

Out[53]:
```
dict_values([['male', 2000], 'male', ['male', 1998], ['Female', 2001]])
```

In [54]:
```
data.iloc[2]
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-54-51f66582218b> in <module>
----> 1 data.iloc[2]

AttributeError: 'dict' object has no attribute 'iloc'
```

In [55]:
```
data['jay']
```

Out[55]:
```
['male', 2000]
```

In [56]:
```
dataitems = list(data.items())
```

In [57]:
```
dataitems
```

Out[57]:
```
[('jay', ['male', 2000]),
 ('kiran', 'male'),
 ('vijay', ['male', 1998]),
 ('Sanskriti', ['Female', 2001])]
```

In [63]:
```
type(dataitems[0][1])
```

Out[63]:
```
list
```

In [61]:
```
len(dataitems[1])
```

Out[61]:
```
2
```

Without using isinstance function

In [66]:
```
count=0
for i in range(len(dataitems)):
    if type(dataitems[i][1]) == list:
            count+=1
print(count)
```

```
3
```

RECURSIVE FUNCTIONS

In [4]:
```
## Factorial with recursion functions

def factorial(n):
    if n == 1:
        return 1
    else:
        return (n*factorial(n-1))

print(factorial(5))
```

```
120
```

In [8]:
```
## Factorial without recursive function

num = int(input("Number for factorial?"))
factorial = 1

for i in range(1,num+1):
    factorial = factorial*i
print(num,'!'," = ",factorial)
```

```
Number for factorial?5
5 !  =  120
```

In [53]:
```
## Find prime numbers:
prime = []
for i in range(2,20):
    c = 0
    for j in range(1,i):
        if i%j == 0:
            c+=1
    if c == 1:
        prime.append(i)
print(prime)
print(sum(prime))
```

```
[2, 3, 5, 7, 11, 13, 17, 19]
77
```

In [49]:
```
## Check if given number is prime or not

n = int(input("Number?"))
for i in range(2,n):
    if n%i == 0:
        print('not a prime number')
        break
    else:
        print('Prime number')
        break
```

```
Number?3
Prime number
```

In [52]:
```
## Palindrome

s = input()
if s == s[::-1]:
    print('Palindrome')

else:
    print('not')
```

```
kiopr
not
```

Write a given string in reverse order.

In [1]:
```
str1 = input()
list1 = str1.split(' ')
print(list1)
list2 = list1[::-1]
print(list2)
str2 = ' '.join(list2)
print(str2)
```

```
Prajyot Tugaonkar
['Prajyot', 'Tugaonkar']
['Tugaonkar', 'Prajyot']
Tugaonkar Prajyot
```

Write unique values from list.

In [2]:
```
mylist = [1,2,2,2,3,3,4,5,5,6,6]
new_list = []
for i in mylist:
    if mylist.count(i) == 1:
        new_list.append(i)

print(new_list)
```

```
[1, 4]
```

In [3]:
```
## Using list comprehension
[i for i in mylist if mylist.count(i) == 1]
```

Out[3]:
```
[1, 4]
```

Find the occurences of each element

In [13]:
```
str1 = "a,a,a,b,b,c,c,c"
list1 = str1.split(',')
count_ele = []

str_ele = []
for ch in list1:
    if not ch in str_ele:
        count_ele.append(f"{ch}:{list1.count(ch)}")
        str_ele.append(ch)
print(",".join(count_ele))
```

```
a:3,b:2,c:3
```

Lambda Functions

In [20]:
```
b = [[6,8],[90,4],[23,67]]
c = [2,8,6,9,10,23,14]
b.sort(key= lambda x:x[0])
print(b)
print(sorted(c))
```

```
[[6, 8], [23, 67], [90, 4]]
[2, 6, 8, 9, 10, 14, 23]
```

In [42]:
```
try:
    n = int(input("How many numbers? "))
except:
    print("Please enter number")
else:
    list_num = []
    try:
        for i in range(n):
            num = int(input())
            list_num.append(num)
        mean = lambda x: sum(x)/len(x)
        print(mean(list_num))
    except:
        print("Some error occured")
```

```
How many numbers? 3
55
120
101
92.0
```

In [38]:
```
2+5+3+6+4
```

Out[38]: 20

In [39]:
```
20/5
```

Out[39]: 4.0

Number Patterns

In [4]:
```
# Increasing Triangle with increasing numbers
n = 5
p = 1
for i in range(n): # This loop goes into rows
    for j in range(i+1): # This loop goes into columns
        print(p,end=' ')#here is space so that there is space bwtween each sta
    p+=1
    print()
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

In [5]:
```
# Increasing Triangle decreasing numbers
n = 5
p = n
for i in range(n): # This loop goes into rows
    for j in range(i+1): # This loop goes into columns
        print(p,end=' ')#here is space so that there is space bwtween each sta
    p-=1
    print()
```

```
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
```

In [6]:
```
# Increasing Triangle Even numbers
n = 5
p = 0
for i in range(n): # This loop goes into rows
    for j in range(i+1): # This loop goes into columns
        print(p,end=' ')#here is space so that there is space bwtween each sta
    p+=2
    print()
```

```
0
2 2
4 4 4
6 6 6 6
8 8 8 8 8
```

In [8]:
```
# Increasing Triangle Even numbers
n = 5
for i in range(n): # This loop goes into rows
    for j in range(i+1): # This loop goes into columns
        if (i%2==0):
            print(1,end=' ')
        else:
            print(2,end=' ')#here is space so that there is space bwtween each
    print()
```

```
1
2 2
1 1 1
2 2 2 2
1 1 1 1 1
```

In [11]:
```
n = 5
p = 1
for i in range(n-1):
    for j in range(i,n):
        print(' ',end=' ')

    for j in range(i+1):
        print(p, end=' ')

    for j in range(i):
        print(p,end=' ')
    p+=1
    print()
for i in range(n):
    for j in range(i+1):
        print(' ', end=' ')
    for j in range(i,n):
        print(p,end=' ')
    for j in range(i,n-1):
        print(p, end=' ')
    p+=1
    print()
```

```
          1
        2 2 2
      3 3 3 3 3
    4 4 4 4 4 4 4
  5 5 5 5 5 5 5 5 5
    6 6 6 6 6 6 6
      7 7 7 7 7
        8 8 8
          9
```

In [14]:
```
n = 5
p = 1
for i in range(n-1):
    for j in range(i,n):
        print(' ',end=' ')

    for j in range(i+1):
        print(p, end=' ')

    for j in range(i):
        print(p,end=' ')
    p+=1
    print()
for i in range(n):
    for j in range(i+1):
        print(' ', end=' ')
    for j in range(i,n):
        print(p,end=' ')
    for j in range(i,n-1):
        print(p, end=' ')
    p-=1
    print()
```

```
          1
        2 2 2
      3 3 3 3 3
    4 4 4 4 4 4 4
  5 5 5 5 5 5 5 5 5
    4 4 4 4 4 4 4
      3 3 3 3 3
        2 2 2
          1
```

In [16]:
```
# Increasing Triangle with increasing numbers
n = 5
for i in range(n): # This loop goes into rows
    p = 1
    for j in range(i+1): # This loop goes into columns
        print(p,end=' ')#here is space so that there is space bwtween each sta
        p+=1
    print()
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

In [20]:
```
n = 5
for i in range(n):
    for j in range(i+1):
        print(' ', end=' ')
    p = 1
    for j in range(i,n):
        print(p,end=' ')
        p+=1
    print()
```

```
  1 2 3 4 5
    1 2 3 4
      1 2 3
        1 2
          1
```

In [22]:
```
# Hill Pattern
n = 5
for i in range(n):
    for j in range(i,n):
        print(' ',end=' ')
    p=1
    for j in range(i+1):
        print(p, end=' ')
        p+=1

    for j in range(i):
        print(p,end=' ')
        p+=1
    print()
```

```
          1
        1 2 3
      1 2 3 4 5
    1 2 3 4 5 6 7
  1 2 3 4 5 6 7 8 9
```

In [23]:
```
n = 5
for i in range(n): # This loop goes into rows
    p = n
    for j in range(i+1): # This loop goes into columns
        print(p,end=' ')#here is space so that there is space bwtween each sta
        p-=1
    print()
```

```
5
5 4
5 4 3
5 4 3 2
5 4 3 2 1
```

In [24]:
```
n = 5
for i in range(n):
    p=n
    for j in range(i+1):
        print(' ', end=' ')
    for j in range(i,n):
        print(p,end=' ')
        p-=1
    print()
```

```
5 4 3 2 1
  5 4 3 2
    5 4 3
      5 4
        5
```

In [25]:
```
n = 5
k = n
for i in range(n):
    p=k
    for j in range(i+1):
        print(' ', end=' ')
    for j in range(i,n):
        print(p,end=' ')
        p-=1
    k-=1
    print()
```

```
5 4 3 2 1
  4 3 2 1
    3 2 1
      2 1
        1
```

In [30]:
```
n = 5
for i in range(n):
    for j in range(i,n):
        print(' ',end = ' ')
    p=1
    for j in range(i):
        print(p,end = ' ')
        p+=1
    for j in range(i+1):
        print(p,end=' ')
        p-=1
    print()
```

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

In [33]:
```
n = 5
p=1
for i in range(n):
    for j in range(i+1):
        print(p,end = ' ')
        p+=1
    print()
```

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

In [ ]:
```

```