In [21]:
```
n=5
for i in range(1,n+1):
    for j in range(1,i+1):
        print(" ",end="")
    for k in range(i,n+1):
        print("*",end=' ')
    print()
```

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

In [ ]:
```
```

In [36]:
```
n=5
for i in range(1,n+1):
    for j in range(1,i+1):
        print(" ",end=" ")
    for k in range(n+1-i,0,-1):
        print(k,end=' ')
    print()
```

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

Q wpp to count total odd and even number also do sum of them. take input n times from user for total number

In [45]:
```
n=int(input("Enter total number count:"))
even=0
odd=0
evensum=0
oddsum=0
while n>0:
    num=int(input(f"Enter number:"))
    if num%2==0:
        even+=1
        evensum=evensum+num
    else:
        odd+=1
        oddsum=oddsum+num
    n=n-1
print('even',even)
print('odd',odd)
print('even_sum',evensum)
print('odd_sum',oddsum)
```

```
Enter total number:4
Enter number:1
Enter number:2
Enter number:3
Enter number:4
even 2
odd 2
even_sum 6
odd_sum 4
```

Q. wpp to compute product of odd digits in given number or 0 if there are not any odd number

In [51]:
```
num=int(input("Enter number:"))
p=1
while num>0:
    r=num%10
    if r%2!=0:
        p=p*r
    num=num//10
if p==1:
    print(0)
else:
    print(p)
```

```
Enter number:567
35
```

Q.wpp to check the given number if disarium number or not

In [55]:
```
import math
num=int(input("Enter number:"))
sum=0
n=num
l=int(math.log10(num))+1  #For length of number
while num>0:
    r=num%10
    sum=sum+r**l
    l=l-1
    num//=10
if sum==n:
    print("Disarium")
else:
    print("not")
```

```
Enter number:175
Disarium
```

Unit 3

Function and scoping

1.Bild in function eg.print,type,input

2.User defined function

```
syntax:-
def function_name(parameter):
    body of function
```

In [64]:
```
def wish(name):
    print("hello",name,"Good Morning")
wish("Arman")
wish("Mitul")
```

```
hello Arman Good Morning
hello Mitul Good Morning
```

In [62]:
```
wish()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-62-c0e1380c0921> in <module>
----> 1 wish()

TypeError: wish() missing 1 required positional argument: 'name'
```

Diffrent Category of UDF(User defined function)

1.function with no perameter and no return type

In [2]:
```
def printline():
    s=input("Enter name:")
    print(s)
printline()
```

```
Enter name:a
a
```

2.function with perameter and no return type

In [3]:
```
def printline(s):
    print(s)
printline("x")
```

```
x
```

3.function with perameter and with return type

In [7]:
```
def printline(s):
    return s
a=printline("x")
print(printline("Mitul"))
print(a)
```

```
Mitul
x
```

3.function with no perameter and with return type

In [1]:
```
def printline():
    s=input("Enter name:")
    return s
print(printline())
```

```
Enter name:mitul
mitul
```

Q.write a function accept n and print odd number beteen 1 to n

In [4]:
```
n=int(input("Enter number:"))
def oddnum(n):
    for i in range(1,n+1,2):
        print(i)
oddnum(n)
```

```
Enter number:9
1
3
5
7
9
```

Return stament

In [6]:
```
def add(x,y):   # function return None
    x+y
result=add(10,20)
```

```
print(result)
print(add(10,20))
```

```
None
None
```

In [9]:
```
def sum_sub(a,b):
    sum=a+b
    sub=a-b
    return sum,sub # its return in tupal imp 10,20,30 is tupal also
x=sum_sub(30,40)
print(x)
a,b=sum_sub(10,20)
print(a)
print(b)
```

```
(70, -10)
30
-10
```

Docstring

what function do

```
Signature: math.sqrt(x, /)
Docstring: Return the square root of x.
Type:      builtin_function_or_method
```

In [12]:
```
def squared_number(x):
    """Argument Passed into x returns square of x"""
    return x*x
t=squared_number(10)
print(t)
print(squared_number.__doc__)
```

```
100
Argument Passed into x returns square of x
```

```
Signature: squared_number(x)
Docstring: Argument Passed into x returns square of x
```

Types of Argument

```
def f1(a,b):
    f1(10,20)


a,b if formal arguments
10,20 is actual argumbet<.pre>
```

1.Positinal argument

In [14]:
```
def sub(a,b):
    print(a-b)
sub(10,20)
sub(20,10)
```

```
-10
10
```

2.keyword argumnet

In [18]:
```
def wish(name,msg):
    print("Hello",name,msg)
wish(name="arman",msg="Good Morning")#keyword argumnet
wish(msg="Good Morning",name="arman")#keyword argumnet

wish("gm","arman") #Positinal argument
```

```
Hello arman Good Morning
Hello arman Good Morning
Hello gm arman
```

In [19]:
```
wish("arman",msg="Good Morning") # positional argument first
```

```
Hello arman Good Morning
```

In [20]:
```
wish(msg="Good Morning","arman")
```

```
  File "<ipython-input-20-9aa17dc9c760>", line 1
    wish(msg="Good Morning","arman")
                            ^
SyntaxError: positional argument follows keyword argument
```

In [21]:
```
wish(name='Arman','GM')
```

```
  File "<ipython-input-21-1291ab64c742>", line 1
    wish(name='Arman','GM')
                      ^
SyntaxError: positional argument follows keyword argument
```

3.Default Argumnet

In [22]:
```
def wish(name='Guest'):
    print("Hello",name)
wish("Arman")
wish()
```

```
Hello Arman
Hello Guest
```

4.Variable Length Argument

In [33]:
```
def sum(*n):
    print(n)
    print(type(n))
    total=0
    for n1 in n:
        total+=n1
    print("total sum:",total)
sum() #  <---
sum(10)
sum(10,20,30)
sum(10,20,30,40,50,60)
```

```
()
<class 'tuple'>
total sum: 0
(10,)
<class 'tuple'>
total sum: 10
(10, 20, 30)
<class 'tuple'>
total sum: 60
(10, 20, 30, 40, 50, 60)
<class 'tuple'>
total sum: 210
```

In [34]:
```
def f1(n1,*s):
    print(n1)
    for i in s:
        print(i)
f1(10)
f1(10,20,30,40)
```

```
10
10
20
30
40
```

In [35]:
```
f1()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-35-b27bf7c7aafe> in <module>
----> 1 f1()

TypeError: f1() missing 1 required positional argument: 'n1'
```

In [36]:
```
def f1(*s,n1):
    print(n1)
    for i in s:
        print(i)
    print(n1)# all arg go in s accsept keyword arg
f1(10)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-36-cb84ed326aec> in <module>
      4         print(i)
      5     print(n1)
----> 6 f1(10)
      7 f1(10,20,30,40)

TypeError: f1() missing 1 required keyword-only argument: 'n1'
```

In [37]:
```
f1(10,20,30,n1=40)
```

```
40
10
20
30
40
```