In [12]:
```
n=int(input("Enter Rows:"))
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()
```

```
Enter Rows:5
  * * * * *
    * * * *
      * * *
        * *
          *
```

In [11]:
```
n=int(input("Enter Rows:"))
for i in range (1,n+1):
    for j in range(1,i+1):
        print(" ",end=" ")
    for k in range(n,i-1,-1):
        print(k,end=" ")
    print()
```

```
Enter Rows:5
  5 4 3 2 1
    5 4 3 2
      5 4 3
        5 4
          5
```

Q.write a program to count toal odd and even numbers and also do the sum of them take a input n from user for the total numbers

In [16]:
```
numbers=int(input("Enter limit:"))
countE=0
countO=0
sumO=0
sumE=0
for i in range(1,numbers+1):
    num=int(input("Enter a number:"))
    if(num%2==0):
        countE+=1
        sumE+=num
    else:
        countO+=1
        sumO+=num
print("Odd count:",countO)
print("Odd sum:",sumO)
print("Even count:",countE)
print("Even sum:",sumE)
```

```
Enter limit:6
Enter a number:1
Enter a number:2
Enter a number:3
Enter a number:4
Enter a number:5
Enter a number:6
Odd count: 3
Odd sum: 9
```

```
Even count: 3
Even sum: 12
```

Q.WAP to compute the product of odd digits in a given number or 0 if there are not any odd numbers

In [25]:
```
number=int(input("Enter a number:"))
temp=number
product=1

count=0
while temp>0:
    digit=temp%10
    if(digit%2!=0):
        product*=digit
        count+=1
    temp//=10
if(count==0):
    print("NO odd digits so product is 0.")
else:
    print("Product:",product)
```

```
Enter a number:2222
NO odd digits so product is 0.
```

Q.WAP to find if the gievn number is Disarium or not

In [37]:
```
n=int(input("Enter a number"))
sum=0
temp=n
length=str(n)
len1=len(length)
while temp>0:
    for i in range(len1,0,-1):
        digit=temp%10
        sum+=digit**i
        temp//=10

if(sum==n):
    print("It is a disarium number")
else:
    print("It is not a disarium number")
```

```
Enter a number112
It is not a disarium number
```

Unit 3:Functions and scoping

1.)Built in function-print,hype,input

2.)User defined function

syntax

```
def function_name(parameters):

    body of function

    return value
```

def(mandatory)

return(optional)

In [40]:
```
def wish(name):
    print("Hello",name,"Good morning")
wish("Arman")
wish("John")
# wish-gives error
```

```
Hello Arman Good morning
Hello John Good morning
```

* Different categories of UDF(User Defined Function)

- 1.)Function with no parameters and no return type

In [44]:
```
def printline():
    s=input("Enter name:")
    print(s)

printline()
```

```
Enter name:kalu
kalu
```

2.)Function with parameters and no return type

In [49]:
```
def printlines(s):
    print(s)

s=input("Enter name:")
printlines(s)
```

```
Enter name:kaliya
kaliya
```

3.)Function with parameters and with return type

In [51]:
```
def printlines(s):
    return s

s=input("Enter name:")
k=printlines(s)
print(k)
```

```
Enter name:sam
sam
```

4.)Function without parameters and with return type

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

```
Enter name:kallu kaliya
kallu kaliya
```

Q.WAF to accept n and print odd numbers between 1 to n

In [3]:
```
def countO(n):
    for i in range(1,n+1):
        if(i%2!=0):
            print(i)

number=int(input("Enter a number"))
countO(number)
```

```
Enter a number50
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
```

In [6]:
```
def add(x,y):
    x+y # returned nothing
result=add(10,20)
print(result)
print(add(20,30))
```

```
None
None
```

In [13]:
```
def sum_sub(a,b):
    sum=a+b
    sub=a-b
    mul=a*b
    dev=a//b
    return sum,sub
x,y=sum_sub(30,40)
```

```
print(x)
print(y)
```

```
70
0
```

In [16]:
```
import math
```

math.# on clicking tab we can get dropdown of the lists

Docstring

In [19]:
```
def square_number(x):
    """Argument passed into x returns square of x"""
    return x*x
t=square_number(5)
print(t)
print(square_number.__doc__)
```

```
25
Argument passed into x returns square of x
```

Types of argument

- def f1(a,b):formal argument
- f1(10,20):actual argument

Positional argument

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

```
-10
10
```

Keyword argument

In [25]:
```
def wish(name,msg):
    print("Hello",name,msg)
wish(name="Arman" ,msg="Good morning")
wish(msg="Good morning", name="Arman")
```

```
Hello Arman Good morning
Hello Arman Good morning
```

In [26]:
```
wish("Arman",msg="Good morning")
wish(name="Arman","Good morning")
```

```
  File "<ipython-input-26-5efad6d102f7>", line 2
    wish(name="Arman","Good morning")
                      ^
SyntaxError: positional argument follows keyword argument
```

Default argument

In [27]:
```
def wish(name="Guest"):
    print("Hello",name)
wish("Arman")
wish()
```

```
Hello Arman
Hello Guest
```

Variable length argument

def f1(*n)

In [34]:
```python
def sum(*n):
    print(n)
    total=0
    for n1 in n:
        total+=n1
    print("The sum is:",total)
sum(10,20,30)
sum(10)
sum()
```

```
(10, 20, 30)
The sum is: 60
(10,)
The sum is: 10
()
The sum is: 0
```

In [40]:
```python
def f1(n1,*s):
    print(n1)
    for i in s:
        print(i)
f1(10)
f1(10,20,30,40)
f1(10,"A","B","C")
```

```
10
10
20
30
40
10
A
B
C
```

In [45]:
```python
def f1(*s,n1):# gives error because no vlaue gets in n1
    for i in s:
        print(i)
    print(n1)
f1(10,n1=20)
f1(10,20,30,n1=40)
```

```
10
20
10
20
30
40
```