🐍

PYTHON

15-04-2024

By: Aaftab Zohra and Yashashwini Sai

Under the guidance of Bijen Singha sir.

Creator of Python - *Dutchman Guido van Rossum*

STRUCTURE

- input
- algorithm
- output

INPUT

they are of 2 types

1. **Run time/dynamic**(if the error occurs during run time it is said to be run time error)
2. **Compile time/static**(if the error occurs due to syntactical errors or spelling mistakes then it is called as compile time error)

- In python if input() - is a default function for string
  n=input()// n stores data in string format
  n==int(input())//n stores data in integer format
- Python doesn't use braces, it uses the concept of indentation
- Colon is used in conditional statements

> *Some important things to remember:*

OPERATORS

1. MEMBERSHIP OPERATOR ⇒in

only works with strings

checks if first thing is present in second

```
#ex 1
if 'hi ' in 'hi,hello':
    print("hi")
else:
    print("bye")
print("abc")

#prints bye abc because 'hi ' as has space in it

#ex 2
if 1 in 123:
       print(1)
elif 1 in '123':
       print(2)
else:
       print(3)

#membership operator only works on strings and not on integers
```

2. IDENTITY OPERATOR ⇒is

checks if both are identical are not

```
if not true is true:
      print(true)
elif not true is false:
       print(false)
else:
       print("error")

#o/p: false
```

3. TERNARY OPERATOR

SYNTAX : <expression>if<if condition>else<else condition>

```
x=y=10
z=1+(x if x>y else y)+2
print(z)

#ans=13



#ex:1
age =20
x 'can vote' if age>18 else 'cannot vote'
print(x)

#o/p:can vote

#ex 2:
x=y==10
z= 1+(x if x>y else y)+2

#o/p:13
```

~Derived Datatypes

List	Tuple	Dictionary	Set
a collection of different datatypes	tuple doesn’t allow the data to be changed it is immutable.	in python dictionary contains key and values	set only shows unique values
empty list - a=[ ]	empty tuple - a=( )	empty dictionary - a={ }	empty set - a=set( )

a[ ]
print(type(a))
o/p:
<class ‘list’>
	a()
	print(type(a))
	o/p:
	<class ‘tuple’>
		a{}
		print(type(a))
		o/p:
		<class ‘dictionary’>
			a=set()
			print(type(a))
			o/p:
			<class ‘set’>

list rep:
a=[”hey”,”h”,3.09,9]
o/p:
[”hey”,”h”,3.09,9]
	tuple rep:
	a=(”hey”,”h”,3.09,9)
	o/p:
	(”hey”,”h”,3.09,9)
			a=”bijen”
			a[0]=p
			print(a)

always represented by index value			#this only prints bijen as string is not mutable
list is mutable			

- just extra

```
#ex 1
if 3 and 0 or 2:
    print("true")
else:
    print("false")

    #prints true

 #ex 2
 if 3>7:
     if -3>-6:
           print("nested if")
     else:
           print("nested else")
else:
       print("else")
#o/p: else

#ex 3
if 11>15 or 12<15:
      if true and false:
             if not false:
                    print(1)
                         if 101 and 543:
                                print(2)
                         else:
                               print(3)
               else:
                      print(4)
```

- all non zero integers will return true in python and only zero will return false

```
if 999:
    print("hi")

   #this prints hi as 999 is a non zero integer
```

- condition is a statement which will return in true or false

```
if print("hello"):
    print("hi")

    #this prints only hello as - if print("hello") is a statement and not
```

```
while print("hi"):
    print("hello")

    #prints hi
```

```
while 999:
    print("hello")

    #prints hello n number of times
    #if while 0: then it prints no output
```

- Python stores variables with same value in the same location
- checking location of a variable : use id()

```
a=1
b=1
print(id(a),id(b))

#prints the location of the variable
```

- else can be used inside loops only (only when there’s no break inside loop i.e. for/while will run completely)

```
for i in range(3):
   print(i,"i am sorry")
   #break(if used prints "0 i am sorry"
else:
 print("not sorry")
```

o/p:

0 i am sorry
1 i am sorry
2 i am sorry
not sorry

- continue will skip the lines below it and go to next statement or loop
- compile time error : syntax or mistake in program
- run time error : error after running
- default datatype - string

basically concatenates ex : n=2 then n+n=22

```
n=int(input())
ans=n+n
print(ans)
```

o/p:
4
8

- python doesn't have concept of braces it uses indentation
- use colon(:) -to represent end of condition

  -after else:

- flag-uses binary digits, used when there are only two conditions

2 MAIN CONCEPTS OF PROGRAMMING

CONDITIONAL

-if: (exactly one condition)

```
#ex1
n=int(input())
if n%2==0:
     b=7
            print(n, b)  #program doesn't execute because of indentation

#ex2
```

```
if n%2==0:

        b=7
        print(n, b)
```

-if else: (exactly 2 conditions)

```
n=int(input())
if n%2==0:
        b=7
        print(n, b)
else:
        print(b)      #doesn't get printed as b is defined in if block not

#o/p: name b is not defined//run time error
```

-**elif**(no limit to no of conditions)

example:

```
n=int(input())
if n>0:
  print("positive")       #(indendation is important in the next statement
elif n<0:
  print("negative")
else:
  print("0")
```

-nested if (all conditions need to be satisfied)

```
if height<5.5
 if weight==65
  if gender==female
    print("All conditions satisfied")
```

-dictionary(switch can be implemented in python using dictionary(switcher.get() is the function used to get the value from switcher)

CONTROL

-for

-

while loop -*while loop doesn't work if it equal to zero or if it starts from zero*

-no do while loop in python

-

break

-

continue

-

pass : has no effect on the program

*programs done using for loop or while loop can also be done using recursion

FOR LOOP

*reduces one by default. if 10 is given it'll run up to 9
SYNTAX : for i in range():

inside range we have ~initialization

~number of times to iterate

~step(basically skipping)

example :

```
n=input()
for i in range(1,10,2):
   print(n)
```

*try (-1,-10,-2) (1,-10,-2) (10,-1,-2) so on...

*in order to get the values of i or know how many times the loop is getting executed we can write the program like this

```
n=input()
for i in range(10):
print(i,n)
```

o/p:

0 i am sorry
1 i am sorry
2 i am sorry
3 i am sorry
4 i am sorry
5 i am sorry
6 i am sorry
7 i am sorry
8 i am sorry
9 i am sorry

Extra examples:

```
n=input()
for i in range(1,10,2):
        print(i, n)
```

//range is 1 to 10 and it'll jump by 2
so output will be
1 sorry
3 sorry
5 sorry
7 sorry
9 sorry

```
n=input()
for i in range(1,10,-2):
         print(i, n)
```

no o/p is generated as range is 1 to 10 but step is -2 which means move behind which can't be done

```
n=input()
for i in range(-10,1,-2):
```

```
          print(i, n)
```

no o/p as -10-2 is -12 which means to move behind
it can't be done

```
n=input()
for i in range(10,-1,-2):
          print(i)
```

o/p
8
6
4
2
0

In python loops can be executed along with conditional statements which cant be done in c

example:

```
#ex1
n=input()
for i in range(3):
      print(i, n)
else:
      print("not sorry")



#ex2
n=input()
for i in range(3):
        print(i ,n)
        break
else:
        print("not sorry")

#o/p: 0 i am sorry
#since break is given else part is not executed

#ex3
n=input()
```

```
for i in range(3):
        print(i)
        continue
        print("hi")
else:
        print("not sorry")
```

ex3 o/p:
0
1
2
not sorry

#after continue the lines which will be written those will be skipped where as break will break the entire loop

> ***Imp:***
> Continue break or pass must be used only inside the loops
> If there is conditional part along with the loop we can't use continue break or pass inside the conditional part

> ***~QUESTIONS***

1. check for leap year

year

if divisible by 4

check if divisible by 100

check if divisible by 400

if satisfies all conditions then LEAP YEAR

```
year=int(input())
if year%4==0:
 if year%100==0:
  if year%400==0:
   print("leap year")
  else:
   print("not leap")
 else:
  print(" leap")
else:
 print("not leap")
```

2. prime number

- prime number is a number which is divisible by itself and 1

ex: 17

17 must not be divisible by any number b/w 1-17

- range starts from 2 because 1 is divisible by all numbers

```
n=int(input())
for i in range(2,n):       #(2,n//2)
if n%i==0:
   print("not prime")
else:
  print("prime")
```

*using flag:

```
n=int(input())
flag=0
for i in range(2,n):
 if n%i==0:
  flag=1
  break
if flag==1:
  print("not prime")
else:
  print("prime")
```

3. Greatest common divisor(GCD)

- min()- determines the smallest number b/w a and b
- g- will be the least gcd i.e.1 because 1 is a divisor for all numbers and will iterate to i
- range lies b/w 1 and smallest number in all cases

ex: 12 - 1 2 3 4 6 12

36 - 1 2 3 6 9 12 18 36

ans=12

```
a=int(input())
b=int(input())
m=min(a,b)
g=1
for i in range(1,m+1):
  if a%i==0 and b%i==0:
    g=i
print(g)
```

4. Least Common Multiple(LCM)

*max()

*g not equal to 1 but bigger number

*limit starts from max no because bigger no can be multiple of small but not vice versa

ex: 6 is multiple of 3

3 is not multiple of 6

```
a=int(input())
b=int(input())
m=max(a,b)
g=m
for i in range(m,(a*b)+1):
    if i%a==0 and i%b==0:
     g=i
     break
print(g)
```

16-04-2024

5. Find reverse of a number

ex: n=1234

→rev=0 stores the result

→divide 1234 by 10 to get remainder as 4 and store it in rem

rem=n%10

→rev=rev*10+rem

→then update the n from 1234 to123 then 12 then 1 so on

n=n//10 this will give us 123.4 but as we used // we ignore the decimal part

```
n=int(input())
rev=0
while n>0:                  #while n!=0
    rem=n%10
    rev=rev*10+rem
    n=n//10
print(rev)
```

6. Find multiplication of all the digits in a number

```
n=int(input())
rev=1
while n>0:
    rem=n%10
    rev=rev*rem
    n=n//10
print(rev)

#ans will be multiplication of all numbers
```

7. Find the addition of all the digits in a number

```
n=int(input())
rev=0
while n>0:
    rem=n%10
    rev=rev+rem
    n=n//10
print(rev)

#ans will be addition of all numbers
```

8. Find if a number is palindrome or not

Palindrome is a number read same forward and backward

→store the number in different variable as n will be updated and end value will be 0

→so to compare it with the reverse number we need the original value

so store n in m

```
n=int(input())
m=n
rev=0
while n>0:
    rem=n%10
    rev=rev*10+rem
    n=n//10

if m==rev:
    print("palindrome")
else:
    print("not palindrome")
```

9. Find the number of digits in a number

```
n=int(input())
r=0
while n>0:
    r=r+1
    n=n//10
print(r)
```

10. Find number of even and odd numbers

```
n=int(input())
even=0
odd=0
while n>0:
 rem=n%10
 if rem%2==0:
    even=even+1
 else:
   odd=odd+1
 n=n//10
print(even)
print(odd)
```

11. Find factorial of a number

```
n=int(input())
fact=1
for i in range(1,n+1):
    fact=fact*i
print(fact)
```

12. Find fibonacci series

0 1 1 2 3 5 8 13 21

→assume 2 numbers 0 and 1

```
n=int(input())
a=0
b=1
print(a,b,end=" ")
for i in range(3,n+1):
    c=a+b
    print(c ,end=" ")
    a=b
    b=c
```

13. PATTERNS (20)

1.

```
n=int(input())
for i in range(0,n):
    for j in range(1,n+1):
        print(j,end=" ")
    print()
```

o/p:

```
1 2 3

1 2 3

1 2 3
```

2.

```
n=int(input())
for i in range(0,n):
    for j in range(1,n+1):
        print(j*2-1,end=" ")
    print()
```

o/p:

```
1 3 5

1 3 5

1 3 5
```

3.

```
n=int(input())
for i in range(1,n+1):
    for j in range(i,n+i):
        print(j,end=" ")
    print()
```

o/p:

```
1 2 3

2 3 4

3 4 5
```

4.

```
n=int(input())
for i in range(0,n):
    for j in range(i,n+i):
        print(chr(65+j),end=" ")
    print()
```

o/p:

```
A B C

B C D

C D E
```

5.

```
n=int(input())
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j,end=" ")
    print()
```

o/p:

1

1 2

1 2 3

6.

```
n=int(input())
for i in range(1,n+1):
    for j in range(i,i+i):           #(i,2*i)
        print(j,end=" ")
    print()
```

o/p:

1

2 3

3 4 5

7.

```
n=int(input())
for i in range(1,n+1):
    for j in range(i,2*i):
        print(j*2-1,end=" ")
    print()
```

o/p:

1

3 5

5 7 9

8.

```
n=int(input())
for i in range(0,n):
    for j in range(n,i,-1):
        print(j,end=" ")
    print()
```

o/p:

```
3 2 1
3 2
3
```

9.

```
n=int(input())
for i in range(0,n):
  if i==0 or i==n-1:
    for j in range(1,n+1):
      print("*",end=" ")
    print()
  else:
   for j in range(0,n):
      if j==0 or j==n-1:
        print("*",end=" ")
      else:
         print(" ",end=" ")
   print()
```

o/p:

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

10.

```
n=int(input())
for i in range(0,n):
  if i==0 or i==n-1 or i==n-1 or i==n-2:
    for j in range(1,n+1):
      print("*",end=" ")
    print()
  else:
   for j in range(0,n):
      if j==0 or j==1 or j==n-1 or j==n-2 :
        print("*",end=" ")
      else:
        print(" ",end=" ")
   print()
```

o/p:

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

11.

```
n=int(input())
for i in range(n):
  if i==0 or i==n-1:
    for j in range(n):
      print("*",end=" ")
    print()
  elif i==n//2:
        for j in range(n):
           if j==0 or j==n//2 or j==n-1:
            print("*",end=" ")
           else:
            print(" ",end=" ")
        print()
  else:
     for j in range(n):
      if j==0 or j==n-1:
        print("*",end=" ")
      else:
        print(" ",end=" ")
     print()
```

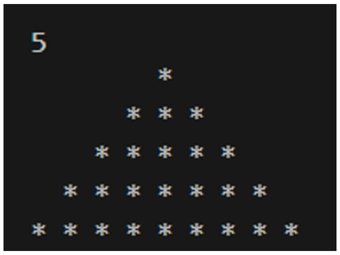
o/p:

```
5
* * * * *
*       *
*   *   *
*       *
* * * * *
```

17-04-2024

12.

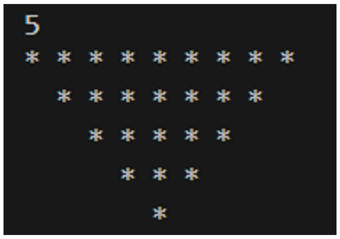
```
n=int(input())
for i in range(1,n+1):
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
```



```
5
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

13.

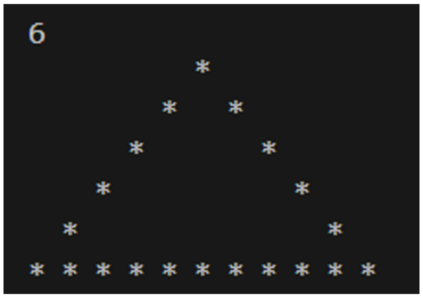
```
n=int(input())
for i in range(n,0,-1):
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
```



```
5
* * * * * * * * *
  * * * * * * *
    * * * * *
      * * *
        *
```

14.

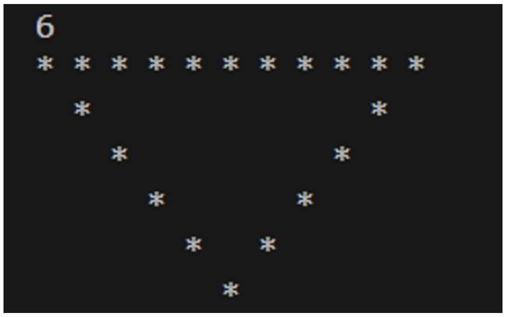
```
n=int(input())
for i in range(1,n+1):
  if i==1 or i==n:
     for j in range(1,n-i+1):
      print(" ",end=" ")
     for j in range(0,2*i-1):
      print("*",end=" ")
     print()

  else:
     for j in range(1,n-i+1):
      print(" ",end=" ")
     for j in range(0,2*i-1):
      if j==0 or j==2*i-2:
       print("*",end=" ")
      else:
        print(" ",end=" ")
     print()
```



```
6
          *
        *   *
      *       *
    *           *
  *               *
* * * * * * * * * * *
```

15.

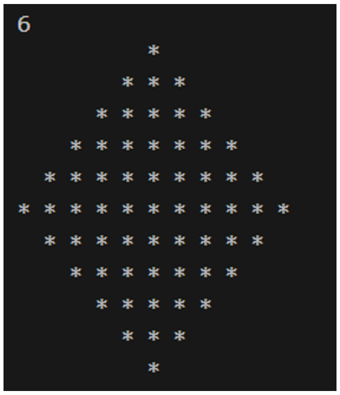
```
n=int(input())
for i in range(n,0,-1):
 if i==0 or i==n:
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
 else:
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
     if j==0 or j==2*i-2:
        print("*",end=" ")
     else:
         print(" ",end=" ")
    print()
```



```
6
* * * * * * * * * * *
  *               *
    *           *
      *       *
        *   *
          *
```

16.

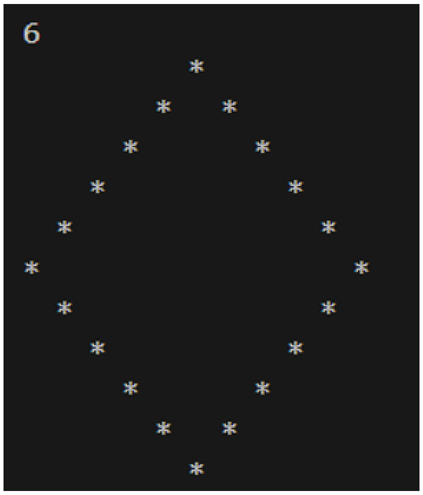
```
n=int(input())
for i in range(1,n+1):
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
for i in range(n-1,0,-1):
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
```



```
6
          *
        * * *
      * * * * *
    * * * * * * *
  * * * * * * * * *
* * * * * * * * * * *
  * * * * * * * * *
    * * * * * * *
      * * * * *
        * * *
          *
```

17.

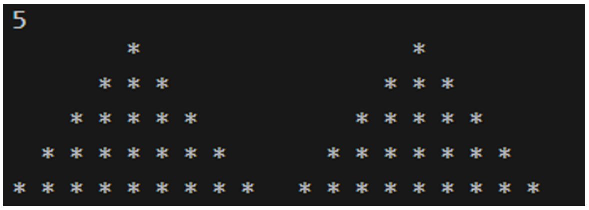
```
n=int(input())
for i in range(1,n+1):
 if i==1:
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
 else:
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
     if j==0 or j==2*i-2:
        print("*",end=" ")
     else:
        print(" ",end=" ")
    print()
for i in range(n-1,0,-1):
 if i==1:
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
 else:
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
     if j==0 or j==2*i-2:
        print("*",end=" ")
     else:
        print(" ",end=" ")
    print()
```



```
6
          *
        *   *
      *       *
    *           *
  *               *
*                   *
  *               *
    *           *
      *       *
        *   *
          *
```

18.

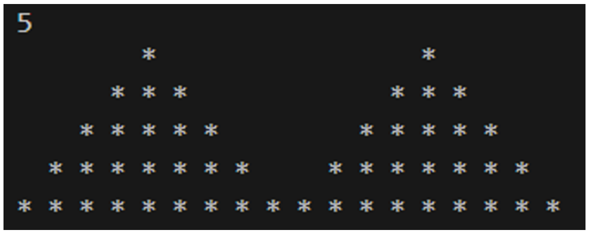
```
n=int(input())
for i in range(1,n+1):
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    for j in range(1,(n-i+1)*2):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
```



```
5
        *                   *
      * * *               * * *
    * * * * *           * * * * *
  * * * * * * *       * * * * * * *
* * * * * * * * *   * * * * * * * * *
```

19.

```
n=int(input())
for i in range(1,n+1):
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    for j in range(1,(n-i+1)*2):
      if j!=(n-i+1)*2-1:
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
```



```
5
        *                 *
      * * *             * * *
    * * * * *         * * * * *
  * * * * * * *     * * * * * * *
* * * * * * * * * * * * * * * * * *
```

20. program to print heart

```
n=int(input())
for i in range(1,n+1):
    for j in range(1,n-i+1):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    for j in range(1,(n-i+1)*2):
      if j!=(n-i+1)*2-1:
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()

for i in range((2*n-1),0,-1):
    for j in range(1,(2*n)-i):
        print(" ",end=" ")
    for j in range(0,2*i-1):
        print("*",end=" ")
    print()
```

a. Program to print K

```
n=int(input())
for i in range(1,n+1):
    for j in range(n-i+1):
      if j==0 or j==n-i:
        print("*",end=" ")
      else:
         print(" ",end=" ")
    print()
for i in range(n,0,-1):
    for j in range(n-i+1):
       if j==0 or j==n-i:
        print("*",end=" ")
       else:
        print(" ",end=" ")
    print()
```

b. Program to print A

```
n=int(input())
for i in range(0,n):
  if i==0 or i==n-1:
    for j in range(1,n+1):
      print("*",end=" ")
    print()
  else:
   for j in range(0,n):
      if j==0:
        print("*",end=" ")
      else:
         print(" ",end=" ")
   print()

for i in range(1,n):
   for j in range(0,n):
      if j==0 or j==n-1:
        print("*",end=" ")
      else:
         print(" ",end=" ")
   print()
```

c. program to print s

```
n=int(input())
for i in range(0,n):
  if i==0 or i==n-1:
    for j in range(1,n+1):
      print("*",end=" ")
    print()
  else:
   for j in range(0,n):
      if j==0:
        print("*",end=" ")
      else:
         print(" ",end=" ")
   print()

for i in range(1,n):
   for j in range(0,n):
      if j==n-1:
        print("*",end=" ")
      elif i==n-1:
       print("*",end=" ")
      else:
         print(" ",end=" ")
   print()
```

d. program to print T

```
n=int(input())
for i in range(0,n):
  if i==0:
    for j in range(1,n+1):
      print("*",end=" ")
    print()
  else:
   for j in range(0,n):
      if j==n//2:
        print("*",end=" ")
      else:
         print(" ",end=" ")
   print()
```

FUNCTION

set of instructions that perform particular task

3 aspects of function are:

1. definition
2. call
3. no declaration

SYNTAX: def function_name():

example:

```
def addition():
    a=3
    b=8
    c=a+b
   print(c)
```

4 TYPES:

1. with parameters and with return values:

```
def addition(a,b):
c=a+b
```

```
return c
addition(5,6)
```

2. with parameters and with no return values:

```
def addition(a,b):
c=a+b
print(c)
addition(5,6)
```

3. without parameters and with return values:

```
def addition():
a=4
b=9
c=a+b
return c
print(addition(c))
```

4. without parameters and without return values:

```
def addition():
a=4
b=9
c=a+b
print(c)
addition()
```

LIST

- reversing a list: (2 ways)

i)

l=[1,2,3,4,5]

print(::-1) { reversing a list by slicing} ii)

```
l=[1,2,3,4,5]
for i in range( len (l)-1,-1,-1):
    print(l[i])
```

i) maximum:

```
l=[10,3,9,1,0,5,60]
m=l[0]
for i in l:
    if i>m:
        m=i
print(m)
```

or

```
l=[10,3,9,1,0,5,60]
m=l[0]
for i in range(len(l)):
    if l[i]>m:
        m=l[i]
print(m)
```

ii) minimum:

```
l=[10,3,9,1,0,5,60]
m=l[0]
for i in range(len(l)):
    if l[i]<m:
        m=l[i]
print(m)
```

or

```
l=[10,3,9,1,0,5,60]
m=l[0]
for i in l:
    if i<m:
        m=i
print(m)
```

-ve indexing					
-6	-5	-4	-3	-2	-1
10	3	0	8	11	1
0	1	2	3	4	5
+ve indexing					

#finding the second largest from the list

```
l=[10,3,9,1,0,5,60]
p=set(l) #set will sort the elements in the list
q=list(p) #as negative indexing can't be done in set we will convert it i
print(q[-2]) #using negative indexing we will find the second largest numb
```

#finding the second smallest from the list

```
l=[10,3,9,1,0,5,60]
p=set(l) #set will sort the elements in the list
q=list(p)
print(q[1]) #using positive indexing we will find the second largest numbe
```

#finding the index value of the 2 numbers which is equal to the target by adding the contents of the list:

```
nums=[1,4,2,6,3,5]
target=6
def addtwo(nums,target):
    a=[]
    for i in range(len(nums)):
        for j in range(i+1,len(nums)):
                if nums[i]+nums[j]==target:
                        a.append(i)
                        a.append(j)
                        return a

print(addtwo(nums,target))
```

o/p:

[0,5]

#finding the index value of the 3 numbers which is equal to the target by adding the contents of the list:

```
nums=[1,4,2,6,3,5]
target=12
def addthree(nums,target):
    a=[]
    for i in range(len(nums)):
        for j in range(i+1,len(nums)):
            for k in range(j+1,len(nums)):
                if nums[i]+nums[j]+nums[k]==target:
                        a.append(i)
                        a.append(j)
                        a.append(k)
                        return a

print(addthree(nums,target))
```

o/p:

[0,3,5]

#remove duplicates from the list(without using set)

```
l=[1,1,1,2,2,3,3,4]
m=[]
for i in l:
    if i not in m:
        m.append(i)
print(m)
```

o/p:

[1,2,3,4]

OR

```
l=[1,1,1,2,2,3,3,4]
m=[]
for i in l:
    if i in m:
        pass
    else:
        m.append(i)
print(m)
```

#remove duplicates from string

```
s="aaaabbbcd"
s1=[]
for i in s:
    if i in s1:
        pass
    else:
        s1.append(i)
print(s1)
```

o/p:

[′a′,′b′,′c′,′d′]

#counting how many times each letter is repeated in a string

```
s="aaaabbbcd"
s1=set(s)
for i in s1:
    count=0
    for j in s:
        if i==j:
            count+=1
    print(i,"-",count)
```

o/p:

d-1

c-1

b-3

a-4

#counting how many times each number is repeated in a string

```
l=[1,3,2,4,1,3,2,2,1,4]
l1=set(l)
for i in l1:
    c=0
    for j in l:
        if i==j:
            c+=1
    print(i,"-",c)
```

o/p:

1-3

2-3

3-2

4-2

#sum of all even and odd numbers in list

```
l=[1,2,3,4,5,6,7,8]
even=0
odd=0
for i in l:
    if i%2==0:
        even+=i
    else:
        odd+=i
```

```
print("even sum=",even)
print("odd sum=",odd)
```

```
even sum=20

odd sum=16
```

#finding common elements from the list

```
l1=[1,3,2,4,5]
l2=[4,6,7,8,9]
l=[]
for i in l1:
    for j in l2:
        if i==j:
            l.append(i)
print(l)
```

OR

```
l1=[1,3,2,4,5]
l2=[4,6,7,8,9]
l=[]
for i in l1:
    if i in l2:
            l.append(i)
print(l)
```

#finding uncommon elements in the list

```
l1=[1,3,2,4,5]
l2=[4,6,7,8,9]
l=[]
for i in l1:
    if i not in l2:
        l.append(i)
for i in l2:
    if i not in l1:
        l.append(i)
print(l)
```

o/p:

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

#finding prime numbers from the list

```
def isprime(x):
    for i in range(2,x//2+1):
        if x%i==0:
            return 0
    else:
        return 1
l=int(input())
u=int(input())
for i in range(l,u+1):
    a=isprime(i)
    if a==1:
        print(i)
```

#converting decimal to binary

```
n=int(input())
b=""
while n>0:
    rem=n%2
    b=b+str(rem)
    n=n//2
print(b[::-1])
```

o/p:

10

1010

#check if a number is a perfect square

```
n=int(input())
i=1
while i*i<n:
    i+=1
if i*i==n:
    print("perfect")
else:
    print("not perfect square")
```

o/p:

6

not perfect square

#check if a number is a power of two

```
n=int(input())
i=1
while 2**i<n:
    i+=1
if 2**i==n:
    print(i)
else:
    print("not power of 2")
```

o/p:

5

not power of 2

8

3

#check if a number is Armstrong number

Armstrong number is a number when each digit of the number powered to the length of the number and added together gives the same number

ex: 371

3^3+7^3+1^3=371

*first find the length of the number i.e. c

*a is to store the result

*the original number is stored in m as it'll be reduced in n=n//10

*later m is compared with a if it is equal then prints Armstrong or else not

```
n=int(input())
c=len(str(n))
```

```
a=0
m=n
while n>0:
    a=a+((n%10)**c)
    n=n//10
if a==m:
    print("armstrong")
else:
    print("not armstrong")
```

#check whether a number is a perfect number

▼ If sum of the divisors of the given number is less than that of the given number, then that number is called as perfect number

▼ If sum of the divisors of the given number is less than that of the given number, then that number is called as perfect number

```
n=int(input())
m=n
sum=0
i=1
while i<n: #for i in range(1,n):
    if n%i==0:
        sum=sum+i
    i+=1
if sum==m:
    print("perfect number")
else:
    print("not perfect number")
```

o/p:

6

perfect number

8

not a perfect number

#check whether the number is abundant number

▼ If the sum of all the divisors of the given number if greater than the number itself. Then that number is called as abundant number

```
n=int(input())
sum=0
i=1
while i<n: #for i in range(1,n):
    if n%i==0:
        sum=sum+i
    i+=1
if sum>n:
    print("abundant number")
else:
    print("not abundant number")
```

o/p:

12

abundant number

6

not abundant number

STRINGS:

converting a list into string:

→split(” “) is used to convert a string into list

→it separates the words in a string with spaces into each element in a list.

→join() is used to convert all the elements of the list into string.

→var=””.join(name of var to be converted)

ex1:

```
s="hey you are idiot"
l=list(s.split(" "))
l=(l[::-1])
s=" ".join(l)
print(s)
```

ex2:

```
s="hey you are idiot"
l=list(s.split(" "))
s=""
for i in l:
    s=s+i[::-1]+" "
print(s)
'''o/p:
yeh ouy era toidi'''
```

#checking if a word is palindrome or not

```
s="Madam"
s1=s.upper()
if s1==s1[::-1]:
    print("palindrome")
else:
    print("not palindrome")

'''o/p:
palindrome'''
```

#check whether the given word is anagram or not

```
a=input()
b=input()
s1=set(a) #to get the unique elements of a
s2=set(b)
if s1==s2:        #checks whether the unique elements are same, if they're
    for i in s1:  #to run the unique elements'''
        if a.count(i)!=b.count(i):
            print("not anagram")
            break
    else:
        print("anagram")
'''o/p:
dog
god
anagram'''
```

RECURSION:

a function which calls itself repeatedly until certain condition is satisfied is called as recursive function.

This process is called as ***recursion***

difference between recursion and loop is:

→a loop will run infinitely
→but by default recursion runs for 0-999 times

a function recurs based on the limit given for recurring.

ex:

```
def recur(x):
    print(x)
    x+=1
    recur(x)
recur()
```

rules for writing recursive func:(format)

1. define function
2. base condition must be given (in 4 to 1, 1 is base condition:1 to 4, 4 is base condition
3. call the function again

#factorial of a number

```
def fact(n):
    if n==1:
        return n
    else:
        return n*fact(n-1)
x=int(input())
print(fact(x))
```

#reverse of a number using recursion

```
def reverse(x,res):
    if x<=0:
        return res
    else:
        rem=x%10
        res=res*10+rem
        return reverse(x//10,res)
    return rev
n=int(input())
rem=0
res=0
print(reverse(n,res))
```

#Fibonacci series using recursion

```
def fib(n):
    if n<=1:
        return n
    else:
        return fib(n-1)+fib(n-2)
x=int(input())
print(fib(x))
```