

```
In [12]: #10,000,000
#1000000
#10a,b
#invalid
#100- 100
n=input("Enter : ")
l=list(n)
err=0
if l[0]=='.' or l[-1]=='.':
    print("Invalid input")
for i in n:
    if i in 'abcdefghijklmnopqrstuvwxyz':
        err+=1
    elif i==',':
        l.remove(',')
    else:
        continue
if err>1 :
    print("Invalid input")
else:
    print(''.join(l))
```

Enter : ,100
Invalid input
100

```
In [13]: # Types of Inheritance
#Single Inheritance
#Multilevel Inheritance
#Hierarchical Inheritance
#Multiple Inheritance
#Hybrid Inheritance
```

```
In [16]: # Single Inheritance

class A: # A is parent class
    name="Mukesh"
    age=36

class B(A): #B(A) means class B can use prop of class A , B is child class
    age=10

obj=B()
#obj.name="Ramesh"
print(obj.age)
print(obj.name)
```

10
Mukesh

```
In [17]: #Multi level Inheritance

class A:
    name="Meher"
    age=18
class B(A):
    age=19
class C(B):
    pass
class D(C):
    age=20

obj=D()
print(obj.name)
```

Meher

```
In [26]: class chairman:
    ch_name="H Babu"
    ch_age=70
class principal(chairman):
    pri_name="Ramesh"
    pri_age=58
class HOD(principal):
    hod_name="Radhika"
    hod_age=48
```

```
obj=HOD()  
print(obj.hod_name)  
print(obj.ch_age)
```

Radhika
70

```
In [28]: class great_grandpa:  
    ggp_name="Satyanarayana Murthy"  
    ggp_age="Past"  
    class grandpa(great_grandpa):  
        gp_name="BalaYogi"  
        gp_age=73  
    class father(grandpa):  
        father_name="MSN Murthy"  
        age=48  
    class daughter(father):  
        daughter_name="Meher Keerthana"  
        age=19  
  
obj=daughter()  
print("Great Grandpa Name : ",obj.ggp_name)  
print("Grandpa Name : ",obj.gp_name)  
print("Father Name : ",obj.father_name)  
print("Daughter Name : ",obj.daughter_name)
```

Great Grandpa Name : Satyanarayana Murthy
Grandpa Name : BalaYogi
Father Name : MSN Murthy
Daughter Name : Meher Keerthana

```
In [31]: # Hierarchical Inheritance  
class Person:  
    name=""  
    gender=""  
    age=""  
    pass  
class Faculty(Person):  
    salary=1000  
    subject=""  
    age=""  
    pass  
class Student(Person):  
    roll_num=""  
    branch="cse"  
    pass  
  
obj=Student()  
print(obj.branch)
```

cse

```
In [36]: # Multiple Inheritance  
class P1:  
    def m1(self):  
        print("In parent class 1")  
class P2:  
    def m1(self):  
        print("In parent class 2")  
#class C(P2,P1):  
#    pass    In parent class 2  
class C(P1,P2):  
    pass  
  
obj=C()  
#obj.m1()  
obj.m1()
```

In parent class 1

```
In [1]: class jewellery_shop:  
    def shopping(self):  
        print("Buy Diamond Necklace")  
        print("Temple gold jewellery")
```

```

class furniture_shop:
    def shopping2(self):
        print("Buy Neutral colors")
        print("Make it look like cozy :)")
class choice(jewellery_shop,furniture_shop):
    pass

obj=choice()
print("Jewellery shopping : ")
obj.shopping()
print()
print("Furniture shopping : ")
obj.shopping2()

```

Jewellery shopping :
Buy Diamond Necklace
Temple gold jewellery

Furniture shopping :
Buy Neutral colors
Make it look like cozy :)

```

In [8]: from random import randint,random
a=randint(1,10)
id=random()*1000
print(id)
print(a)

```

454.2444811982386
10

```

In [15]: import random
print("Select rock,paper,scissors")
print()
player1=input("Player1 : ")
player2=random.choice(['rock', 'paper', 'scissors'])
print("PPlayer 2 :",player2)

p1=0
p2=0
if player1=='rock':
    if player2=='scissors':
        p1+=1
    elif player2=='paper':
        p2+=1
    else:
        pass
elif player1=='paper':
    if player2=='scissors':
        p2+=1
    elif player2=='rock':
        p1+=1
    else:
        pass
else:
    if player2=='rock':
        p2+=1
    elif player2=='paper':
        p1+=1
    else:
        pass

if p1>0:
    print("Player1 won the game")
elif p2>0:
    print("PPlayer2 won the game")
else:
    print("Tie , Try Again!")

```

Select rock,paper,scissors

Player1 : rock
Player 2 : scissors
Player1 won the game

```

In [17]: # method overriding is run-time polymorphism
class A:
    def method_1(self,a,b):

```

```

        print("Sum of 2 numbers : ",a+b)
class B(A):
    def method_1(self,a,b):
        print("Mul of 2 numbers : ",a*b)
    def method_1(self,abc):
        print("value is : ",abc)
obj=B()
obj.method_1(10,20)

```

```

-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_23556\321594710.py in <module>
      9         print("value is : ",abc)
     10     obj=B()
--> 11 obj.method_1(10,20)

TypeError: method_1() takes 2 positional arguments but 3 were given

```

In [33]:

```

num1=int(input("Enter : "))
num2=int(input("Enter : "))
num1=bin(num1)
num1=list(num1[2:])
num2=bin(num2)
num2=list(num2[2:])

for i in range(len(num1)):
    if num1[i]=='1':
        num1[i]='0'
    else:
        num1[i]='1'
for i in range(len(num2)):
    if num2[i]=='1':
        num2[i]='0'
    else:
        num2[i]='1'

print()
a=''.join(num1)
b=''.join(num2)
c=int(a,2)
d=int(b,2)
print(c^d)

```

Enter : 5
Enter : 7

2

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js