Number system conversion

- binary base(0,1),and divide by number /2 & count in reverse order,octal base :(0,7)
- hexadecimal:base (0,9 and a=10,b=11,c=13...till f,(A,F/a,f)
- In real time we use in ip config..When we want to know ip address

```
In [1]: 25
Out[1]: 25

In [2]: bin(25)
Out[2]: '0b11001'

In [3]: 0b11001
Out[3]: 25

In [4]: bin(45)
Out[4]: '0b101101'

In [6]: int(0b101101)
Out[6]: 45

In [7]: oct(13)
Out[7]: '0o15'

In [8]: oct(67)
Out[8]: '0o103'

In [9]: int(0o103)
Out[9]: 67

In [10]: hex(6)
Out[10]: '0x6'

In [11]: hex(70)
Out[11]: '0x46'

In [12]: hex(10)
```

```
Out[12]:  '0xa'
```

```
In [13]:  int(0x43)
```

```
Out[13]:  67
```

```
In [14]:  0xa
```

```
Out[14]:  10
```

Swap variable between two numbers in different methods

```
In [21]:  x=5
          y=4
```

```
In [22]:  x,y=y,x
```

```
In [23]:  x
```

```
Out[23]:  4
```

```
In [24]:  y
```

```
Out[24]:  5
```

```
In [25]:  x1=67
          x2=43
```

```
In [26]:  temp=x1
          x1=x2
          x2=temp
```

```
In [28]:  print(x1)
          print(x2)
```

```
43
67
```

```
In [30]:  # using addition,sub method
          a=67
          b=45
```

```
In [31]:  a=a+b
          b=a-b
          a=a-b
          print(a)
          print(b)
```

```
45
67
```

In [35]:

```
a1=10
b1=20
```

In [36]:

```
a1=a1+b1
b1=a1-b1
a1=a1-b1
print(a1)
print(b1)
```

```
20
10
```

In [38]:

```
print(0b101)
print(0b110)
```

```
5
6
```

In [39]:

```
print(bin(11))
print(0b1011)
```

```
0b1011
11
```

In [43]:

```
a=7
b=8
```

In [44]:

```
a=a^b
b=a^b
a=a^b
print(a)
print(b)
```

```
8
7
```

Bitwise oparator

1. Compliment(~)
2. And(&)
3. OR(|)
4. XOR(^)
5. Left shift(<<)
6. Right shift(>>)

In [1]:

```
print(bin(12))
print(bin(13))
```

```
0b1100
0b1101
```

In [2]:

```
# COMPLEMENT(~) (TILED OR TILD)
~12 # why we get -13.first we undestand what is complement means(reverse of binary
```

Out[2]:

```
-13
```

In [3]:

```
~1
```

Out[3]:

```
-2
```

In [4]:

```
~45
```

Out[4]:

```
-46
```

In [5]:

```
~2
```

Out[5]:

```
-3
```

BITWISE OPERATOR

- bit wise and operator
- AND-LOGICAL OPERATOR|||&-BITWISE AND OPERATOR
- (we know that 1&1 is 1)
- 12-00001100
- 13-00001101
- when we are add both then outut we will get as 12

In [6]:

```
12&13
```

Out[6]:

```
12
```

In [7]:

```
12&13
```

Out[7]:

```
12
```

In [8]:

```
1&1
```

Out[8]:

```
1
```

In [9]:

```
1|1
```

Out[9]:

```
1
```

In [10]:

```
1&0
```

Out[10]:

```
0
```

In [11]:
```
# in XOR if the both number are different then we will get 1 or else we will get
12^13
```

Out[11]:
```
1
```

In [12]:
```
25^30
```

Out[12]:
```
7
```

In [13]:
```
bin(25)
```

Out[13]:
```
'0b11001'
```

In [14]:
```
bin(30)
```

Out[14]:
```
'0b11110'
```

In [15]:
```
int(0b11110)
```

Out[15]:
```
30
```

BIT WISE LEFT OPERATOR

- bitwise left operator bydefault you will take 2 zeros()
- 10 biunary oparator is 1010|also i can say 1010

In [16]:
```
10<<2
```

Out[16]:
```
40
```

In [17]:
```
50<<3
```

Out[17]:
```
400
```

BITWISE RIGHT SHIFT OPERATOR

In [20]:
```
10>>2
```

Out[20]:
```
2
```

In [21]:
```
bin(20)
```

Out[21]:
```
'0b10100'
```

In [22]:
```
50>>2
```

Out[22]:
```
12
```

Import math module

```
In [24]: x=sqrt(625) # here sqrt is inbuid fuc
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[24], line 1
----> 1 x=sqrt(625)

NameError: name 'sqrt' is not defined
```

```
In [27]: import math # importing math module
```

```
In [28]: x=math.sqrt(625)
         x
```

```
Out[28]: 25.0
```

```
In [30]: print(math.floor(2.9)) #minimum or least value
```

```
2
```

```
In [31]: print(math.ceil(2.9)) # maximum or highest value
```

```
3
```

```
In [32]: print(math.pow(6,2))
```

```
36.0
```

```
In [33]: print(math.pi) #constant value
```

```
3.141592653589793
```

```
In [34]: print(math.e)
```

```
2.718281828459045
```

```
In [35]: import math as m
         m.sqrt(1225)
```

```
Out[35]: 35.0
```

```
In [36]: import math as m
         m.pow(2,3)
```

```
Out[36]: 8.0
```

```
In [37]: from math import pow
         pow(2,3)
```

```
Out[37]: 8.0
```

In [39]:
```
from math import ceil
ceil(8.97)
```

Out[39]:
```
9
```

In [40]:
```
from math import *
print(pow(4,6))
print(ceil(5.5))
```

```
4096.0
6
```

In [42]:
```
round(pow(9,2))
```

Out[42]:
```
81
```

user input function||comand line input

In [1]:
```
r=input()
z=input()
c=r+z
print(c)
```

```
23
```

In [2]:
```
z1=input('first number')
z2=input('second number')
z3=z1+z2
z3
```

Out[2]:
```
'87'
```

In [3]:
```
type(z1)
type(z3)
```

Out[3]:
```
str
```

In [4]:
```
x1=input("Enter number")
a=int(x1)
x2=input("Enter a number")
b=int(x2)
c=a+b
c
```

Out[4]:
```
30
```

In [6]:
```
a=int(input("1st number"))
b=int(input("2nd numbber"))
c=a+b
c
```

Out[6]:
```
55
```

```
In [7]: ch=input("Enter a char")
        print(ch)

        raghava

In [8]: print(ch[0])

        r

In [9]: print(ch[-1])

        a

In [10]: ch=input("Enter charecter")[0]
         ch

Out[10]: 'nani'

In [11]: ch

Out[11]: 'nani'

In [1]: ch=input("Enter")[1:3]
        ch

Out[1]: 'an'

In [2]: ch=input("Enter")
        ch

Out[2]: '54'
```

Eval fuc using input

```
In [3]: result=eval(input('enter a expr'))
        print(result)

        50

In [ ]:
```