

Photo Album

by sumit kumar

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2+3
5
>>> a=55
>>> type(a)
<class 'int'>
>>> a='bharti'
>>> type(a)
<class 'str'>
>>> a=4.5
>>> type(a)
<class 'float'>
>>> loosely| coupled
```

```
>>> a
```

```
4.5
```

```
>>> c=int(a)
```

```
>>> c
```

```
4
```

```
>>> a=88
```

```
>>> b=float(a)
```

```
>>> b
```

```
88.0
```

```
>>> a,b=5,6
>>> a+b
11
>>> a-b
-1
>>> a*b
30
>>> 2**3
8
>>> a/b
0.8333333333333334
>>> a//b
0
>>> a%b
5
>>> _+15
20
```

```
>>> a/b
0.8333333333333333
>>> a//b
0
>>> a%b
5
>>> _+15
20
>>> c=a>b
>>> c
False
>>> int(c)
0
>>> c=a<b
>>> int(c)
1
>>> a==b
False
>>> a!=b
True
```

```
>>> a,b=5,6
>>> a>b
False
>>> int(a>b)
0
>>> int(a<b)
1
>>> a<b
True
>>> a>b and a<b
False
>>> a>b or a<b
True
>>> not(a>b and a<b)
True
```

```
>>> int(a<b)  
1  
>>> a<b  
True  
>>> a>b and a<b  
False  
>>> a>b or a<b  
True  
>>> not(a>b and a<b)  
True  
>>> a=a+b  
>>> a  
11  
>>> a+=b  
>>> a  
17  
>>> a & b  
0
```

$a = 17 = 10001_2$

$b = 6 = 00110_2$

$a \& b$

$a | b$

The diagram shows a handwritten binary division problem. The dividend is 17 (written as 10001), and the divisor is 6 (written as 00110). The quotient is calculated as 2 (circled in green with the number 23) and the remainder is 5 (circled in green with the number 0).

$$\begin{array}{c} A \oplus B \\ \bar{A}B + A\bar{B} \end{array}$$

A	B	$A \wedge B$
0	0	0
0	1	1
1	0	1
1	1	0

I

$a = \Gamma = 10\%$

$b = 6 = 0.110$

10111

$a \wedge b$

```
>>> a|b
```

```
23
```

```
>>> a^b'
```

SyntaxError: EOL while scanning str

```
>>> a^b
```

```
23
```

```
>>> ~a
```

```
-18
```

```
>>> a=-7
```

```
>>> ~a
```

```
6
```

```
>>> a=5
```

```
>>> a<<2
```

a**^**b

-**(n+1)**

-**(-7+1)**

;(-+6)

```
>>> a|b
```

```
23
```

```
>>> a^b'
```

```
SyntaxError: EOL while scanning string literal
```

```
>>> a^b
```

```
23
```

```
>>> ~a
```

```
-18
```

```
>>> a=-7
```

```
>>> ~a
```

```
6
```

```
>>> a=5
```

```
>>> a<<2
```

```
20
```

```
>>> a>>2
```



```
>>> ab=['bharti',5.6,2]
>>> type(ab)
<class 'list'>
>>> cd=[1,2,3,4,6,7,8,9,1,2,3,4,5,6,7,8,9,1,2,3,4,5,6]
>>> cd[0]
1
>>> cd[5]
7
>>> cd[-1]
6
>>> cd[3:8]
[4, 6, 7, 8, 9]
>>> cd[4:]
[6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6]
>>> cd[:8]
[1, 2, 3, 4, 6, 7, 8, 9]
>>>
```

```
>>> min(cd)
1
>>> max(cd)
9
>>> sum(cd)
106
>>> cd.append(99)
>>> cd
[1, 2, 3, 4, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 99]
>>> cd.insert(5,666)
>>> cd
[1, 2, 3, 4, 6, 666, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 99]
```

```
>>> cd.extend(ab)
>>> cd
[1, 2, 3, 4, 6, 666, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 99, 'bha
rti', 5.6, 2]
>>> cd.pop()
2
>>> cd
[1, 2, 3, 4, 6, 666, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 99, 'bha
rti', 5.6]
>>> cd.pop(6)
7
>>> cd
[1, 2, 3, 4, 6, 666, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 99, 'bharti
', 5.6]
>>> cd.remove(666)
>>> cd
[1, 2, 3, 4, 6, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 99, 'bharti', 5.
6]
>>>
```



```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AM  
D64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> ab=(1,2,3,8,7,6,5,4,9,8,1,2,3,4,4)  
>>> type(ab)  
<class 'tuple'>  
>>> ab[0]  
1  
>>> ab[4]  
7  
>>> ab[4]=77  
Traceback (most recent call last):  
  File "<pyshell#4>", line 1, in <module>  
    ab[4]=77  
TypeError: 'tuple' object does not support item assignment  
>>> list is mutable
```

```
>>> ab.count(2)
2
>>> ab.count(4)
3
>>> ab.index(2)
1
>>> len(ab)
15
>>> min(ab)
1
>>> max(ab)
9
>>> sum(ab)
67
```

```
>>> ab[1:4]
```

```
(2, 3, 8)
```

```
>>> A={1,2,3,4}
```

```
>>> B={3,4,5,6}
```

```
>>> A.difference(B)
```

```
{1, 2}
```

```
>>> B.difference(A)
```

```
{5, 6}
```

```
>>> A.symmetric_difference(B)
{1, 2, 5, 6}
>>> A
{1, 2, 3, 4}
>>> B
{3, 4, 5, 6}
>>> A.difference_update(B)
>>> A
{1, 2}
>>> B
{3, 4, 5, 6}
>>> A={1,2,3,4}
>>> A.symmetric_difference_update(B)
>>> A
{1, 2, 5, 6}
>>> A={1,2,3,4}
>>> A
{1, 2, 3, 4}
>>> B
{3, 4, 5, 6}
```

```
>>> A={1,2,3,4}
>>> A.symmetric_difference_update(B)
>>> A
{1, 2, 5, 6}
>>> A={1,2,3,4}
>>> A
{1, 2, 3, 4}
>>> B
{3, 4, 5, 6}
>>> A.intersection(B)
{3, 4}
>>> A
{1, 2, 3, 4}
>>> B
{3, 4, 5, 6}
>>> A.intersection_update(B)
>>> A
{3, 4}
>>> B
{3, 4, 5, 6}
```

A = {1, 2, 3, 4, 5, 6}
B = {3, 4, 5, 6}



```
>>> A.intersection_update(B)
>>> A
{3, 4}
>>> B
{3, 4, 5, 6}
>>> A
{3, 4}
>>> B
{3, 4, 5, 6}
>>> A.issubset(B)
True
>>> B.issubset(A)
False
>>> A.issuperset(B)
False
>>> B.issuperset(A)
True
```

A = {3, 4} B = {3, 4, 5, 6}

6

```
>>> B
{3, 4, 5, 6}
>>> A
{3, 4}
>>> B.pop()
3
>>> B[1]
Traceback (most recent call last):
  File "<pyshell#44>", line 1, in <module>
    B[1]
TypeError: 'set' object is not subscriptable
>>> b
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)
)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ab=[1,2,3,4,8,9,7,6,5,4,3,2,5,2,3,6,8,84]
>>> ab.count(2)
3
>>> len(ab)
18
>>> ab.index(4)
3
>>> ab.reverse()
>>> ab
[84, 8, 6, 3, 2, 5, 2, 3, 4, 5, 6, 7, 9, 8, 4, 3, 2, 1]
>>> ab.sort()
>>> ab
[1, 2, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 8, 8, 9, 84]
>>> ab.reverse()
>>> ab
[84, 9, 8, 8, 7, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2, 1]
```

```
>>> cd
[84, 9, 8, 8, 7, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2, 1]
>>> ef=ab
>>> ef
[84, 9, 8, 8, 7, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2, 1]
>>> id(ab)
1633637175104
>>> id(cd)
1633640777536
>>> if(ef)
SyntaxError: invalid syntax
>>> id(ef)
1633637175104
>>> ab.clear()
>>> ab
[]
>>> cd
[84, 9, 8, 8, 7, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2, 1]
>>> ef
[]
```

```
>>> cd[4]  
7  
>>> cd  
[84, 9, 8, 8, 7, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2, 1]  
>>> cd[4]=66  
>>> cd  
[84, 9, 8, 8, 66, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2, 1]  
>>> cd  
[84, 9, 8, 8, 66, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2, 1]  
>>> type(cd)  
<class 'list'>  
>>> cd.pop()  
1  
>>> cd  
[84, 9, 8, 8, 66, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2]
```

```
>>> cd.pop(4)
66
>>> cd
[84, 9, 8, 8, 6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 2]
...>
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a={1,2,3,4}
>>> b={3,4,5,6}
>>> a.union(b)
{1, 2, 3, 4, 5, 6}
>>> a
{1, 2, 3, 4}
>>> a.update(b)
>>> a
{1, 2, 3, 4, 5, 6}
>>> a.pop()
1
>>> a
{2, 3, 4, 5, 6}
>>> a.remove(4)
>>> a
{2, 3, 5, 6}
>>>
```

```
>>> a={1,2}
>>> b={3,4,5,6}
>>> a.isdisjoint(b)
True
>>> a={1,2,3}
>>> a.isdisjoint(b)
False
>>> a.discard(2)
>>> a
{1, 3}
>>> a={1,2,3,4}
```

```
>>> a.discard(2)
>>> a
{1, 3, 4}
>>> a.remove(2)
Traceback (most recent call last):
  File "<pyshell#20>", line 1, in <module>
    a.remove(2)
KeyError: 2
>>> a.remove(3)
>>> a
{1, 4}
>>> a.discard(3)
>>> a
{1, 4}
```

```
>>> a.add(22)
>>> a
{1, 4, 22}
>>> a.add(5)
>>> a
{1, 4, 5, 22}
>>> b=a.copy()
>>> a
{1, 4, 5, 22}
>>> b
{1, 4, 5, 22}
>>> c=a
>>> c
{1, 4, 5, 22}
>>> id(a)
2976111206176
>>> id(b)
2976111205504
```

```
>>> id(c)
2976111206176
>>> a.clear()
>>> a
set()
>>> c
set()
>>> b
{1, 4, 5, 22}
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a=[1,2,3,4]
>>> a=[1,2,3,4,5,2,3,1]
>>> a
[1, 2, 3, 4, 5, 2, 3, 1]
>>> type(a)
<class 'list'>
>>> a[3]=22
>>> a
[1, 2, 3, 22, 5, 2, 3, 1]
>>> ab=(1,2,3,4,5,6)
>>> ab[3]=22
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    ab[3]=22
TypeError: 'tuple' object does not support item assignment
>>> immutable
```

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> a=['a',6,9.6]
>>> a[0]
'a'
>>> a[1]
6
>>> b=(1,'r',9.8)
>>> b[0]
1
>>> b[1]
'r'
>>>
```

```
a=5
```

```
print(a)
```

Type "help", "copyright", "credits" or "license" for more information.

>>>

===== RESTART: D:/python ba

5

>>>

```
a,b=3,4
if(a>b):
    print("a is greater")
else:
    print('b is greater|')
```

```
>>>  
===== RESTART: D:/py  
5  
>>>  
===== RESTART: D:/py  
b is greater  
>>>
```

```
a=input('enter first number : ')
b=input('enter second number : ')

if(a>b):
    print("a is greater")
else:
    print('b is greater')
```

```
===== RESTART: D:/pyth
enter first number : 55
enter second number : 34
a is greater
>>>
===== RESTART: D:/pyth
enter first number : asdf
enter second number : bhgf
b is greater
```

```
a=input('enter first number : ')
b=input('enter second number : ')
print(a+b)
'''if(a>b):
    print("a is greater")
else:
    print('b is greater')'''
```

```
===== RESTART: D:/py
enter first number : 4
enter second number : 6
46
>>>
===== RESTART: D:/py
enter first number : bharti
enter second number : madaan
bhartimadaan
```

```
a=input('enter first number : ')
b=input('enter second number : ')
c=int(a)
d=int(b)
print(c+d)
'''if(a>b):
    print("a is greater")
else:
    print('b is greater')'''
```

```
===== RESTART: D:/python batch/ifelsedata.py =====
enter first number : 3
enter second number : 4
7
>>>
===== RESTART: D:/python batch/ifelsedata.py =====
enter first number : bharti
enter second number : madaan
Traceback (most recent call last):
  File "D:/python batch/ifelsedata.py", line 3, in <module>
    c=int(a)
ValueError: invalid literal for int() with base 10: 'bharti'
```

```
c=int(input('enter first number : '))
d=int(input('enter second number : '))
print(c+d)
'''if(a>b) :
    print("a is greater")
else:
    print('b is greater')'''
```

```
===== RESTART
```

```
enter first number : 2
```

```
enter second number : 3
```

```
5
```

```
\\
```

```
a=int(input('enter first number : '))
b=int(input('enter second number : '))
if(a>b):
    print("a is greater")
else:
    print('b is greater')
```

```
===== RESTART: 1
enter first number : 4
enter second number : 5
b is greater
>>>
```

```
===== RESTART: 1
enter first number : 5
enter second number : 3
a is greater
>>>
```

```
yob=int(input('Enter year of birth : '))
age=2021-yob
print('age is : ',age)
if(age>=18) :
    print('eligible for voting')
else:
    print('not eligible for voting')
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python batch/agecalc.py =====
Enter year of birth : 1999
age is : 22
eligible for voting
>>>
```

```
a=int(input('Enter first number : '))#8
b=int(input('Enter second number : '))#71
c=int(input('Enter third number : '))#634
if(a>b):
    if(a>c):
        print('a is greatest')
    else:
        print('c is greatest')
else:
    if(b>c):
        print('b is greatest')
    else:
        print('c is greatest')
```

```
Enter first number : 4
Enter second number : 3
Enter third number : 1
a is greatest
>>>
=====
RESTART: D:/python batch/agecalc.py =
Enter first number : 3
Enter second number : 4
Enter third number : 5
c is greatest
>>>
=====
RESTART: D:/python batch/agecalc.py =
Enter first number : 4
Enter second number : 3
Enter third number : 5
c is greatest
>>>
=====
RESTART: D:/python batch/agecalc.py =
Enter first number : 3
Enter second number : 6
Enter third number : 2
b is greatest
```

```
a=int(input('Enter first number : '))#8
b=int(input('Enter second number : '))#7
c=int(input('Enter third number : '))#6
if(a>b and a>c):
    print('a is greatest')
elif(b>c and b>a):
    print('b is greatest')
else:
    print('c is greatest')
```

```
===== RESTART: D:/p
Enter first number : 3
Enter second number : 4
Enter third number : 6
c is greatest
```

```
english=int(input('Enter marks in english : '))
hindi=int(input('Enter marks in hindi : '))
math=int(input('Enter marks in math : '))
ss=int(input('Enter marks in SS : '))
sci=int(input('Enter marks in Science : '))
total=english+hindi+math+ss+sci
avg=total/5
if(avg>=90):
    print('A+')
elif(avg >=80 and avg<90):
    print('A')
elif(avg >=70 and avg<80):
    print('B+')
elif(avg >=60 and avg<70):
    print('B')
elif(avg >=50 and avg<60):
    print('C')
else:
    print('Fail')
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>>
```

```
===== RESTART: D:/python batch/average.py =====
```

```
Enter marks in english : 92
```

```
Enter marks in hindi : 92
```

```
Enter marks in math : 92
```

```
Enter marks in SS : 92
```

```
Enter marks in Science : 92
```

```
A+
```

```
>>>
```

```
===== RESTART: D:/python batch/average.py =====
```

```
Enter marks in english : 80
```

```
Enter marks in hindi : 80
```

```
Enter marks in math : 80
```

```
Enter marks in SS : 80
```

```
Enter marks in Science : 80
```

```
A
```

```
>>> |
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> range(10)
range(0, 10)
>>> ab=list(range(10))
>>> ab
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
>>> ab=list(range(1,10))
>>> ab
[1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> ab=list(range(1,10,2))
>>> ab
[1, 3, 5, 7, 9]
>>> ab=list(range(0,10,2))
>>> ab
[0, 2, 4, 6, 8]
```

```
>>> ab=list(range(10,0))
>>> ab
[]
>>> ab=list(range(10,0,-1))
>>> ab
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
for x in range(10):  
    print(x)
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AM  
D64)] on win32
```

```
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>>
```

```
===== RESTART: D:/python batch/average.py =====
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
>>>
```

```
for x in range(1,11):  
    print(x)
```

1
2
3
4
5
6
7
8
9
10
>>>

```
PM>>> for x in range(1,11,2):
    print(x)
```



1

3

5

7

9

```
a,b=5,9
print('before swapping : a is : ',a,'and b is : ',b)
a=a+b #a=14
b=a-b #14-9 b=5
a=a-b #14-5=a=9

print('after swapping : a is : ',a,'and b is : ',b)
```

```
===== RESTART: C:/Users/sumit kumar/Desktop/swap.py =====
before swapping : a is : 5 and b is : 9
after swapping : a is : 9 and b is : 5
>>>
```

```
a,b=0,1
t=int(input('enter no of terms : '))
#print(a, '\n', b)
for i in range(1, t+1):
    c=a+b
    print(a)
    a=b
    b=c
```

===== RESTART: C:/Users/sumit kumar/Desktop/

enter no of terms : 15

0

1

1

2

3

5

8

13

21

34

55

89

144

233

377

```
'''for i in range(1,10):
    print(i)'''

i=1
while (i<=10):
    print(i)
    i=i+1 #i+=1
```

RESTART : C:/Users/

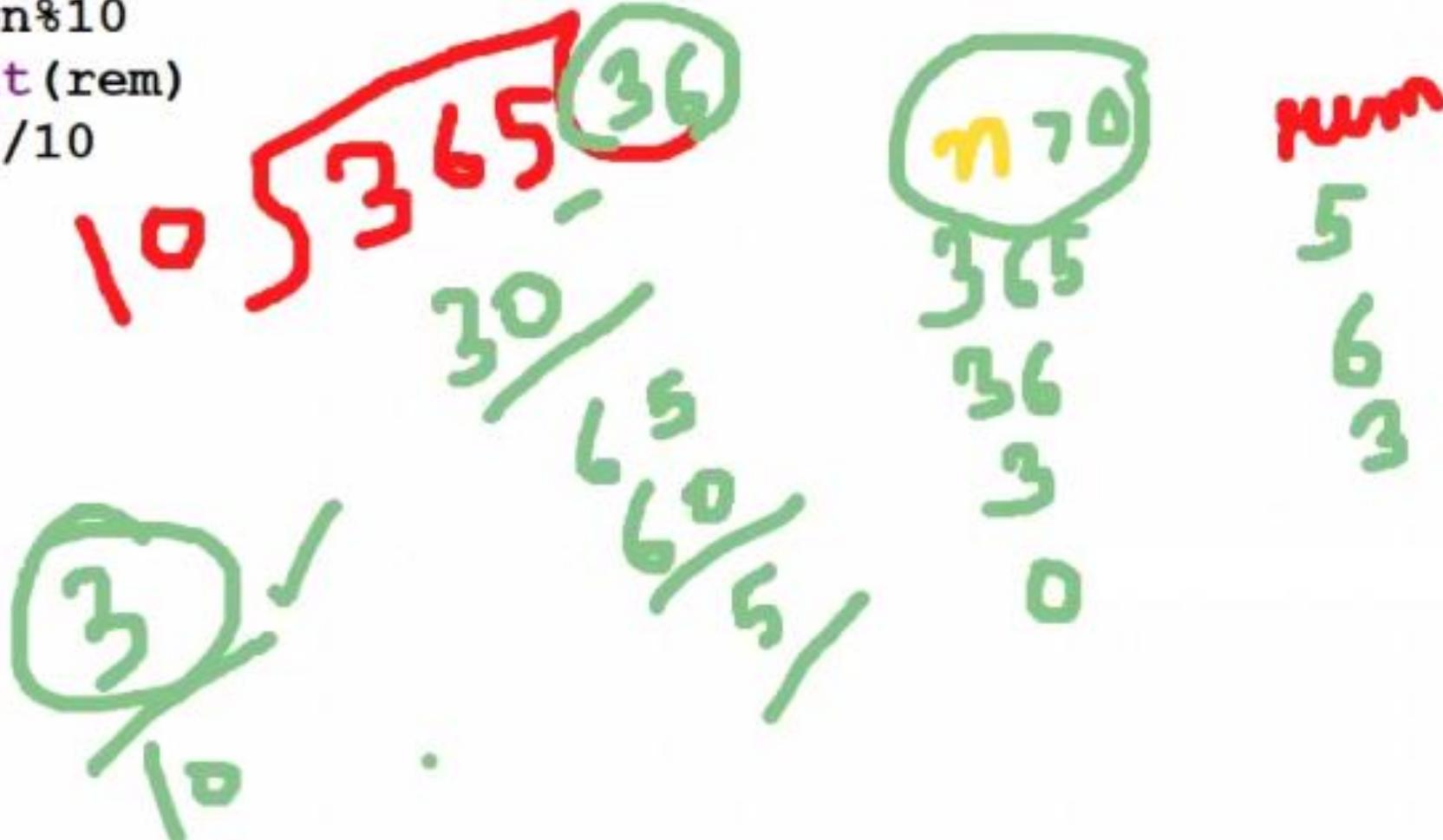
1
2
3
4
5
6
7
8
9
10

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>> n=36  
>>> n=365  
>>> n%10  
5  
>>> n//10  
36  
>>> |
```

```
n=int(input('enter number : '))
while(n>0):
    rem=n%10
    print(rem)
    n=n//10
```



RESULT

enter number : 472

2
7
4

```

t=0
n=int(input('enter number : '))
while(n>0):
    rem=n%10
    t=t*10+rem
    n=n//10
print(t)

```

$$\begin{array}{r}
 105365 \\
 \times 3 \\
 \hline
 36
 \end{array}$$

$$\begin{array}{r}
 365 \\
 \times 3 \\
 \hline
 0
 \end{array}$$

$$\begin{array}{r}
 56 \\
 \times 3 \\
 \hline
 168
 \end{array}$$

$t = 0$
 $t = 0 + 6 \times 10^0$
 $t = 0 + 6 \times 10^0 + 8 \times 10^1$
 $t = 0 + 6 \times 10^0 + 8 \times 10^1 + 1 \times 10^2$
 $t = 0 + 6 \times 10^0 + 8 \times 10^1 + 1 \times 10^2 + 6 \times 10^3$

enter number : 563

3

36

365

```
t=0  
n=int(input('enter number : '))  
while(n>0):  
    rem=n%10  
    t=t*10+rem  
    n=n//10  
print(t)
```

```
===== RESTART: C:/Users/...  
enter number : 563  
365  
>>> |
```

```
t=0
n=int(input('enter number : '))
temp=n
while(n>0):
    rem=n%10
    t=t*10+rem
    n=n//10
print(t)
if(temp==t):
    print('palindrome no')
else:
    print('not a palindrome no')
'''n      rem      t
123  3      3
12   2      32
1    1      321
0'''
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52  
[GCC 9.3.0] on win32  
Type "help", "copyright", "credits" or "license()" for m  
>>>  
===== RESTART: D:\python batch\reverseum  
enter number : 123  
321  
not a palindrome no  
>>>  
===== RESTART: D:\python batch\reverseum  
enter number : 1221  
1221  
palindrome no  
>>>  
===== RESTART: D:\python batch\reverseum  
enter number : 121  
121  
palindrome no  
>>>
```

```
t=0
n=int(input('enter number : '))
temp=n
while(n>0):
    rem=n%10
    t=t+rem**3
    n=n//10
print(t)
if(temp==t):
    print('armstrong no')
else:
    print('not a armstrong no')
'''n   rem   t=0
153   3       27
15     5       125+27=152
1      1       152+1=153
0'''
```

```
===== RESTART: C:\Users\  
enter number : 370  
370  
armstrong no  
>>>  
===== RESTART: C:\Users\  
enter number : 153  
153  
armstrong no  
>>>  
===== RESTART: C:\Users\  
enter number : 555  
375  
not a armstrong no
```

```
n=int(input('enter number : '))
while (n>0) :
    rem=n%10
    print(rem)
    n=n//10
```

RESTART

enter number : 123

3

2

1

>>>

```
n=int(input('enter number : '))
while(n>0):
    rem=n%10
    #print(rem)
    fact=1
    for i in range(1,rem+1):
        fact=fact*i
    print(fact)
    n=n//10
```

===== RESTART:

enter number : 145

120

24

1

>>> |

```
n=int(input('enter number : '))
sum=0
while(n>0):
    rem=n%10
    #print(rem)
    fact=1
    for i in range(1,rem+1):
        fact=fact*i
    #print(fact)
    sum=sum+fact
    n=n//10
print(sum)
```

KERSHAW

center number : 145

145

```
n=int(input('enter number : '))
sum=0
temp=n
while(n>0):
    rem=n%10
    #print(rem)
    fact=1
    for i in range(1,rem+1):
        fact=fact*i
    #print(fact)
    sum=sum+fact
    n=n//10
if(temp==sum):
    print('krishnamurti no')
else:
    print('not a krishnamurti no')
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (A  
D64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: D:/advance python/tkinter/naman data/abcd.py =====  
enter number : 145  
krishnamurti no  
>>>  
===== RESTART: D:/advance python/tkinter/naman data/abcd.py =====  
enter number : 144  
not a krishnamurti no  
>>>  
===== RESTART: D:/advance python/tkinter/naman data/abcd.py =====  
enter number : 144  
49  
not a krishnamurti no  
>>> |
```

```
import array as ar    I  
ar.array()
```

```
import array as ar
from array import *
array()
```

TypeCode	C Type	Python Type	Min. size in bytes
'b'	signed char	int	1
'B'	unsigned char	int	1
'u'	Py_UNICODE	Unicode character	2
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int	int	2
'I'	unsigned int	int	2
'l'	signed long	int	4
'L'	unsigned long	int	4
'f'	float	float	4
'd'	double	float	8

```
>>> import array as ar
>>> ar.array('i',[10,20,30,40,50,60])
array('i', [10, 20, 30, 40, 50, 60])
>>> ab=ar.array('i',[10,20,30,40,50,60])
>>> ab
array('i', [10, 20, 30, 40, 50, 60])
>>> ab[0]
10
>>> ab[1]
20
>>> ab[-1]
60
```

```
>>> ab=ar.array('i',[10,20,30,40,-50,60])
>>> ab
array('i', [10, 20, 30, 40, -50, 60])
>>> ab=ar.array('I',[10,20,30,40,-50,60])
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    ab=ar.array('I',[10,20,30,40,-50,60])
OverflowError: can't convert negative value to unsigned int
```

```
import array as ar
ab=ar.array('i',[1,2,3,4,5,6])
length=len(ab)
for i in range(length):
    print(ab[i])
```

1 |

2 |

3 |

4 |

5 |

6 |

```
import array as ar
ab=ar.array('i',[10,20,30,4,5,6])
for i in ab:
    print(i)
```

10

20

30

4

5

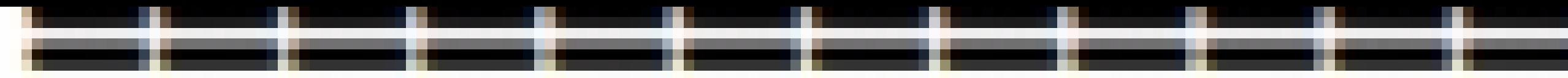
6

```
data=[4,3,1,8,2]
mini=data[0]
for i in range(len(data)):
    if(mini>data[i]):
        mini=data[i]
mini data[i]
4 > 4
4 > 3
3 > 1
1>8
1>2|
```

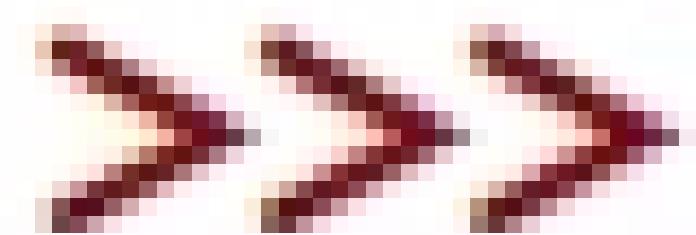
```
data=[4,3,1,8,2]
mini=data[0]
for i in range(len(data)):
    if(mini>data[i]):
        mini=data[i]
print(mini)
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:2  
)] on win32  
Type "help", "copyright", "credits" or "license()" f  
>>>  
===== RESTART: D:/python batch/mini  
4  
3  
1  
1  
1  
>>>
```

```
data=[4,3,1,8,2]
mini=data[0]
for i in range(len(data)):
    if(mini>data[i]):
        mini=data[i]
print(mini)
```



1



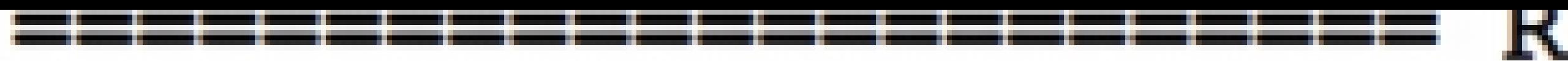
```
data=[4,3,1,8,2]
mini=data[0]
for i in range(len(data)):
    if(mini>data[i]):
        mini=data[i]
print("minimum number is : ",mini)
```

REST

minimum number is : 1



```
import array as arr
data=arr.array('i',[4,3,1,8,2])
mini=data[0]
for i in range(len(data)):
    if(mini>data[i]):
        mini=data[i]
print("minimum number is : ",mini)
```



minimum number is : 1

.....

```
import array as arr
data=arr.array('i',[4,3,1,8,2])
max=data[0]
for i in range(len(data)):
    if(max<data[i]):
        max=data[i]
print("maximum number is : ",max)
```

RE

maximum number is : 8

ANS

```
import array as arr
data=arr.array('i',[4,3,1,8,2])
num=int(input('enter number which u want to search : '))
for i in range(len(data)):
    if(num==data[i]):
        print('found')
    else:
        print('not found')
```

```
----- RESTART: D:/python ba
```

```
enter number which u want to search : 3
```

```
not found
```

```
found
```

```
not found
```

```
not found
```

```
not found
```

```
flag=0 #flag=0 means not found
import array as arr
data=arr.array('i',[4,3,1,8,2])
num=int(input('enter number which u want to search : '))
for i in range(len(data)):
    if(num==data[i]):
        flag=1
if(flag==1):
    print('found')
else:
    print('not found')
```

```
===== RESTART: D:/python batch/min:  
enter number which u want to search : 4  
found  
>>>  
===== RESTART: D:/python batch/min:  
enter number which u want to search : 7  
not found  
>>>  
===== RESTART: D:/python batch/min:  
enter number which u want to search : 3  
found
```

```
flag=0 #flag=0 means not found
import array as arr
data=arr.array('i',[4,3,4,4,2])
num=int(input('enter number which u want to search : '))
for i in range(len(data)):
    if(num==data[i]):
        flag+=1 #flag=flag+1
if(flag>0):
    print('no. found',flag,"times")
else:
    print('not found')
```

num data {
4 == 4
4 == 3
4 == 4
4 == 1
4 == 2
3

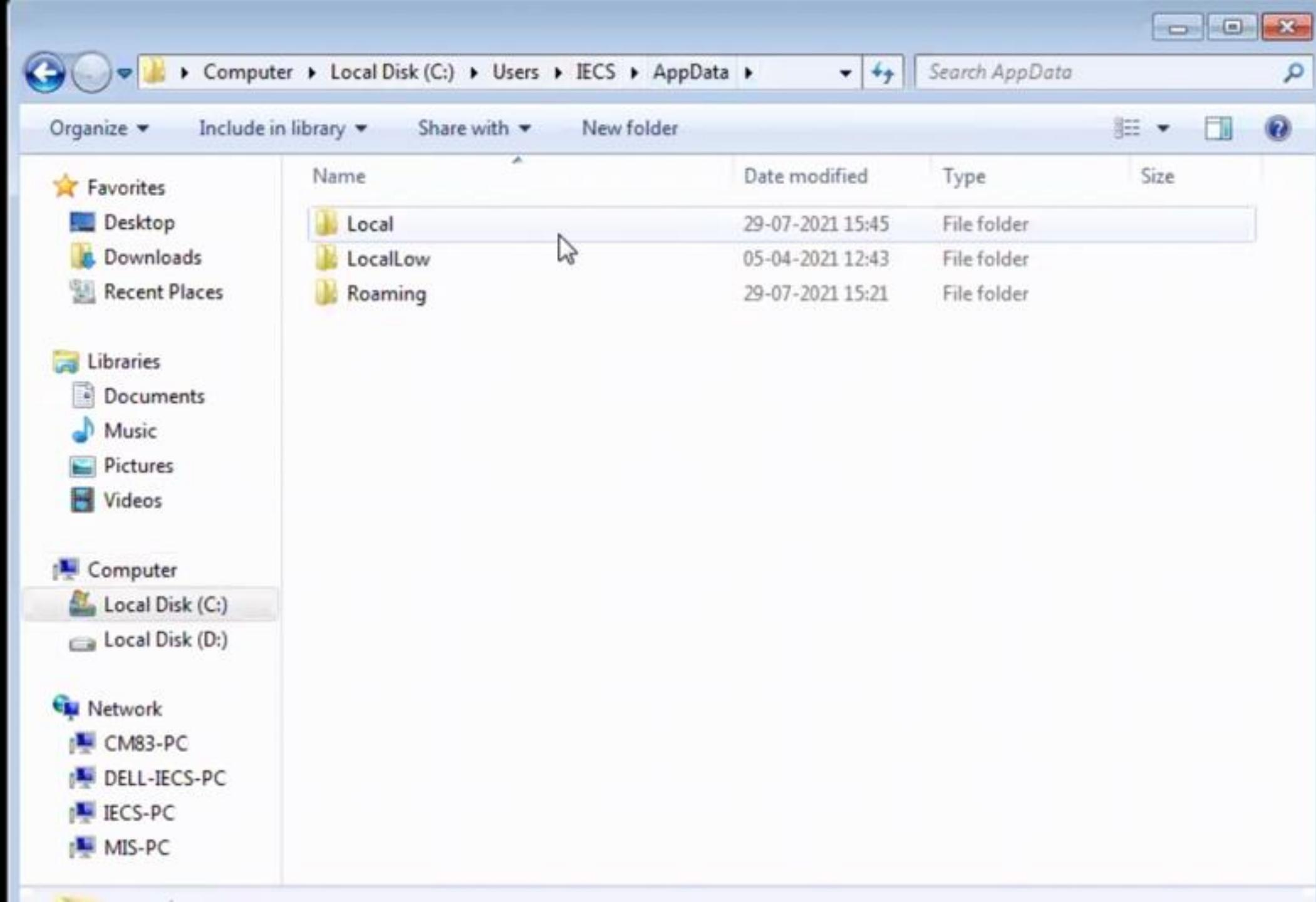
```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
===== RESTART: C:/Users/sumit kumar/Desktop/New folder/flag.c =====
enter number which u want to search : 4
no. found 3 times
>>>
===== RESTART: C:/Users/sumit kumar/Desktop/New folder/flag.c =====
enter number which u want to search : 1
not found
>>>
===== RESTART: C:/Users/sumit kumar/Desktop/New folder/flag.c =====
enter number which u want to search : 2
no. found 1 times
>>>
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import numpy
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    import numpy
ModuleNotFoundError: No module named 'numpy'
>>>
```

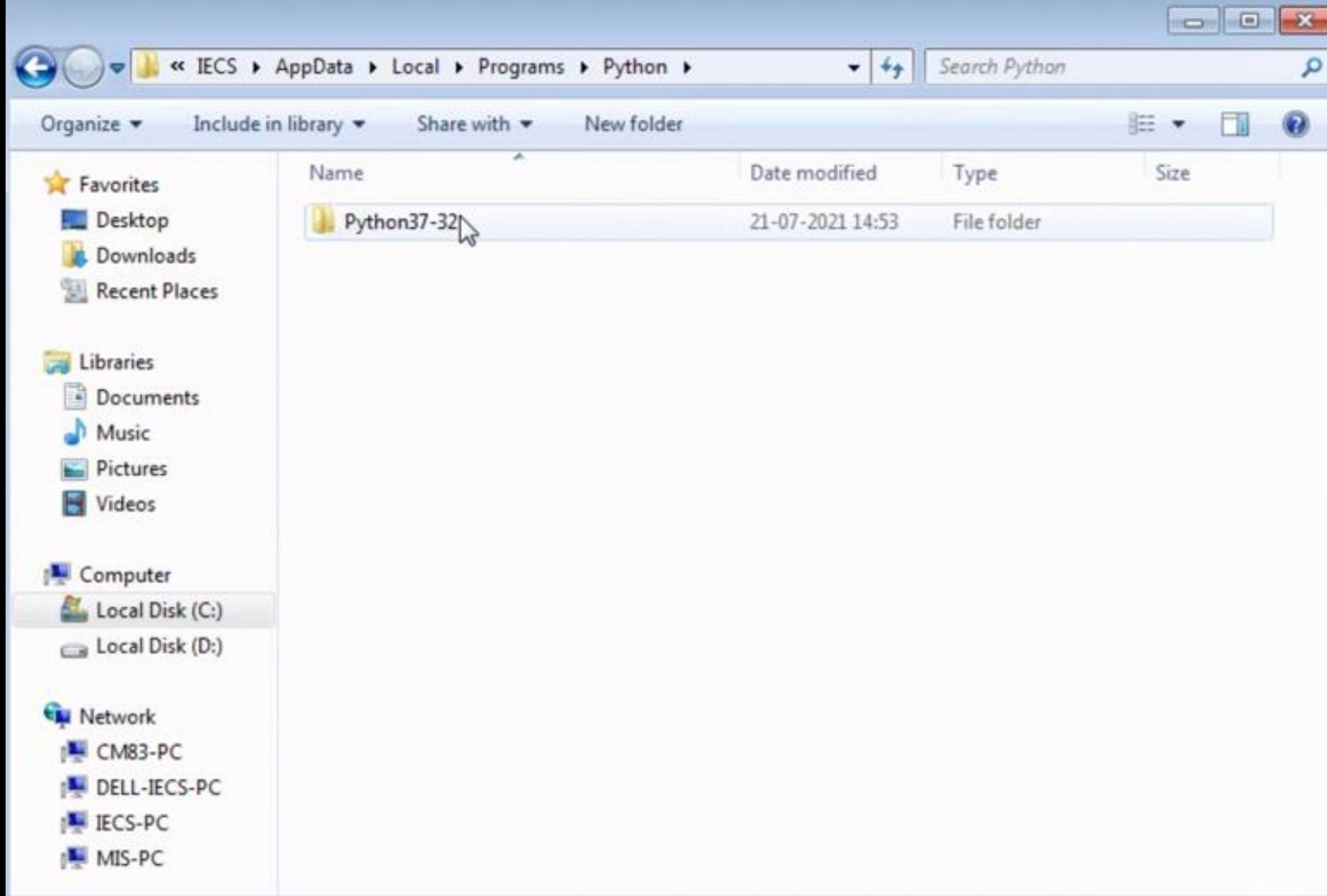
C:\Windows\system32\cmd.exe

Microsoft Windows [Version 6.1.7601]
Copyright © 2009 Microsoft Corporation. All rights reserved.

C:\Users\IECS>pip3 install numpy



Local Disk (C:) > Users > IECS > AppData > Local				
Organize	Include in library	Share with	New folder	
★ Favorites				
Desktop				
Downloads				
Recent Places				
Libraries				
Documents				
Music				
Pictures				
Videos				
Computer				
Local Disk (C:)				
Local Disk (D:)				
Network				
CM83-PC				
DELL-IECS-PC				
IECS-PC				
MIS-PC				
	Name	Date modified	Type	Size
	Arduino15	15-04-2021 10:18	File folder	
	Diagnostics	21-07-2021 12:57	File folder	
	ElevatedDiagnostics	29-07-2021 09:34	File folder	
	eqsoft	05-04-2021 12:43	File folder	
	Google	06-04-2021 15:27	File folder	
	Microsoft	23-07-2021 14:47	File folder	
	Microsoft Help	05-04-2021 11:58	File folder	
	Package Cache	13-04-2021 13:03	File folder	
	pip	29-07-2021 15:45	File folder	
	Programs	05-04-2021 11:55	File folder	
	Temp	29-07-2021 15:45	File folder	
	VirtualStore	03-04-2021 15:21	File folder	
	GDIFFONTCACHEV1.DAT	07-04-2021 09:35	DAT File	107 KB



A screenshot of a Windows File Explorer window. The title bar shows the path: C:\Users\IECS\AppData\Local\Programs\Python\Python37-32\Scripts. The window displays a list of files in this directory, including easy_install, easy_install-3.7, f2py, pip, pip3.7, and pip3. All files are application files (91 KB) and were modified on 05-04-2021 at 11:56. The left sidebar shows Favorites (Desktop, Downloads, Recent Places), Libraries (Documents, Music, Pictures, Videos), Computer (Local Disk (C:), Local Disk (D:)), and Network (CM83-PC, DELL-IECS-PC, IECS-PC, MIS-PC). The status bar at the bottom indicates 6 items.

	Name	Date modified	Type	Size
	easy_install	05-04-2021 11:56	Application	91 KB
	easy_install-3.7	05-04-2021 11:56	Application	91 KB
	f2py	29-07-2021 15:45	Application	91 KB
	pip	05-04-2021 11:56	Application	91 KB
	pip3.7	05-04-2021 11:56	Application	91 KB
	pip3	05-04-2021 11:56	Application	91 KB

Control Panel (2)

 Edit environment variables for your account

 Edit the system environment variables



 See more results

envi

x



Shut down



System Properties

[Computer Name](#)[Hardware](#)[Advanced](#)[System Protection](#)[Remote](#)

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

[Settings...](#)

User Profiles

Desktop settings related to your logon

[Settings...](#)

Startup and Recovery

System startup, system failure, and debugging information

[Settings...](#)[Environment Variables...](#)[OK](#)[Cancel](#)[Apply](#)

Environment Variables



User variables for IECS

Variable	Value
PATH	C:\Users\IECS\AppData\Local\Programs...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

New...Edit...Delete

System variables

Variable	Value
NUMBER_OF_P...	2
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

New...Edit...Delete

Environment Variables



Edit System Variable



Variable name: Path

Variable value: %SystemRoot%\System32\WindowsPowerShell\v1.0\

OK

Cancel

System variables

Variable	Value
NUMBER_OF_P...	2
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\...;
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

New...

Edit

Delete

• • • • • OK Cancel

Environment Variables



Edit System Variable



Variable name: Path

Variable value: c:\Programs\Python\Python37-32\Scripts;

OK

Cancel

System variables

Variable	Value
NUMBER_OF_P...	2
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

New...

Edit...

Delete

OK

Cancel

Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

```
>>> import numpy
```

```
>>> |
```

```
from numpy import *
arr = array([1,4,5,6,7])
print(arr)
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed1  
tel) ] on win32  
Type "help", "copyright", "cre  
>>>  
===== RESTART: C:/Users/IE  
[1 4 5 6 7]  
>>> |
```

```
from numpy import *
arr = array([1,4,5,6,7])
print(arr.dtype)
```

I

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar  
tel) on win32
```

```
Type "help", "copyright", "credits"
```

```
>>>
```

```
===== RESTART: C:/Users/IECS/Des  
[1 4 5 6 7]
```

```
>>>
```

```
===== RESTART: C:/Users/IECS/Des  
int32
```

```
>>> |
```

```
from numpy import *
arr = array([1, 4, 5, 6.8], 7]
print(arr.dtype)
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25  
tel) ] on win32  
Type "help", "copyright", "credits" or '  
>>>  
===== RESTART: C:/Users/IECS/Desktop  
[1 4 5 6 7]  
>>>  
===== RESTART: C:/Users/IECS/Desktop  
int32  
>>>  
===== RESTART: C:/Users/IECS/Desktop  
float64  
>>> |
```

```
from numpy import *
arr = array([1, 4, -5, 6.8, 7])
print(arr.dtype)
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25  
tel) ] on win32  
Type "help", "copyright", "credits" or  
>>>  
===== RESTART: C:/Users/IECS/Desktop/I  
[1 4 5 6 7]  
>>>  
===== RESTART: C:/Users/IECS/Desktop/I  
int32  
>>>  
===== RESTART: C:/Users/IECS/Desktop/I  
float64  
>>>  
===== RESTART: C:/Users/IECS/Desktop/I  
float64
```

```
from numpy import *
arr = linspace(1,2,5)
print(arr)
```

I

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25  
tel) on win32  
Type "help", "copyright", "credits" or '  
>>>  
===== RESTART: C:/Users/IECS/Desktop  
[1 4 5 6 7]  
>>>  
===== RESTART: C:/Users/IECS/Desktop  
int32  
>>>  
===== RESTART: C:/Users/IECS/Desktop  
float64  
>>>  
===== RESTART: C:/Users/IECS/Desktop  
float64  
>>>  
===== RESTART: C:/Users/IECS/Desktop  
[1. 1.25 1.5 1.75 2. ]
```

```
from numpy import *
arr = linspace(1, 20, 5)
print(arr)
```

I

```
===== RESTART: C:/Users/IECS/Desktop.py
[1 4 5 6 7]
>>>
===== RESTART: C:/Users/IECS/Desktop.py
int32
>>>
===== RESTART: C:/Users/IECS/Desktop.py
float64
>>>
===== RESTART: C:/Users/IECS/Desktop.py
float64
>>>
===== RESTART: C:/Users/IECS/Desktop.py
[1. 1.25 1.5 1.75 2. ]
>>>
===== RESTART: C:/Users/IECS/Desktop.py
[ 1. 5.75 10.5 15.25 20. ]
```

```
from numpy import *
arr = linspace(1, 25, 5)
print(arr)
```



```
===== RESTART: C:/Users  
[ 1. 7. 13. 19. 25.]
```

```
from numpy import *
arr = linspace(1, 25, 35)
print(arr)
```

```
===== RESTART: C:/Users/IECS/Desktop/python batch/numydata1.py =====
[ 1.  7. 13. 19. 25.]
>>>
===== RESTART: C:/Users/IECS/Desktop/python batch/numydata1.py =====
[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14. 15. 16. 17. 18.
 19. 20. 21. 22. 23. 24. 25.] . . . . . . . .
```

```
from numpy import *
arr = arange(1L, 25, 25)
print(arr)
```

A horizontal sequence of colored blocks representing a genome or sequence. The sequence consists of two main strands: a top strand with a repeating pattern of blue, orange, and black segments, and a bottom strand with a more complex, multi-colored pattern of blue, orange, yellow, red, purple, and black segments. The strands are separated by a gap.

[1]

A small cluster of colored blocks at the bottom left, consisting of several small squares in yellow, red, purple, and brown.

```
from numpy import *
arr = logspace(1, 40, 5)
print(arr)
```

```
===== RESTART: C:/Users/IECS/Desktop/python batch/numpydata1.py ===
[ 1  3  5  7  9 11 13 15 17 19 21 23]
>>>
===== RESTART: C:/Users/IECS/Desktop/python batch/numpydata1.py ===
[1.00000000e+01 5.62341325e+10 3.16227766e+20 1.77827941e+30
 1.00000000e+40]
```

```
from numpy import *
arr = zeros(5)
print(arr)
```

```
===== RESTART: C:/Users/  
[1.0000000e+01 5.62341325e  
1.0000000e+40]  
>>>  
===== RESTART: C:/Users/  
[0. 0. 0. 0. 0.]
```

```
from numpy import *
arr = ones(5)
print(arr)
```

```
===== RESTART: C:\>
[0. 0. 0. 0. 0.]
>>>
===== RESTART: C:\>
[1. 1. 1. 1. 1.]
```

```
from numpy import *
arr = ones(5,int)
print(arr)
```

```
===== RESTART:
```

```
[1. 1. 1. 1. 1.]
```

```
>>>
```

```
===== RESTART:
```

```
[1 1 1 1 1]
```

```
import numpy as np
ab=np.array([1,6,8,4,3])
print(ab)
no=int(input('enter value : '))
ab=np.append(ab,no)
print(ab)
```

```
----- RESTART: C:/Users/sumit kumar/Desktop/New folder/numpy do append.py -----
[1 6 8 4 3]
enter value : 99
[ 1 6 8 4 3 99]
>>> |
```

```
import numpy as np
ab=np.array([])
print(ab)
size=int(input('enter size of array : '))
for i in range(size):
    data=int(input('enter value'))
    ab=np.append(ab,data)
print(ab)
```

```
----- RESTART: C:/Users/Santosh Kumar/Desktop/New folder/main.py -----
[]

enter size of array : 5
enter value 12
enter value 5
enter value 6
enter value 6
enter value 6
[12.  5.  6.  6.  6.]
```

```
>>> import numpy as np
>>> ab=np.arange(1,20,3)
>>> print(ab)
[ 1  4  7 10 13 16 19]
>>> ab=np.arange(1,24,3)
>>> print(ab)
[ 1  4  7 10 13 16 19 22]
>>> print(ab.shape())
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    print(ab.shape())
TypeError: 'tuple' object is not callable
>>> ab.shape=(2, 4)
>>> print(ab)
[[ 1  4  7 10]
 [13 16 19 22]]
>>> ab.shape=(4, 2)
>>> print(ab)
[[ 1  4]
 [ 7 10]
 [13 16]
 [19 22]]
```

```
>>> ab=np.arange(24)
>>> cd=ab.reshape(2, 4, 3)
>>> print(cd)
[[[0 1 2]
  [3 4 5]
  [6 7 8]
  [9 10 11]]]
```

```
[[12 13 14]
 [15 16 17]
 [18 19 20]
 [21 22 23]]]
```

```
import numpy as np  
ab=np.zeros(6,int)  
print(ab)
```

```
Python 3.8.6 (tags/v3.8.6:db4  
AMD64) ] on win32  
Type "help", "copyright", "cr  
>>>  
===== RESTART: C:/Users  
[0 0 0 0 0 0]  
>>> |
```

```
import numpy as np
r1=int(input('enter number of rows : '))
c1=int(input('enter number of cols : '))
ab=np.zeros([r1,c1],int)| I
print(ab)
```

```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 13:54:46)
[PyPy 6.3.1 (6.3.1+appveyor)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>>
```

I

```
===== RESTART: C:/Users/IECS/Desktop/p1.py =====
```

```
[0 0 0 0 0 0]
```

```
>>>
```

```
===== RESTART: C:/Users/IECS/Desktop/p1.py =====
```

```
enter number of rows : 3
```

```
enter number of cols : 3
```

```
[[0 0 0]
```

```
[0 0 0]
```

```
[0 0 0]]
```

```
>>> |
```

```
import numpy as np
r1=int(input('enter number of rows : '))
c1=int(input('enter number of cols : '))
ab=np.zeros([r1,c1],int)
for r in range(r1):
    for c in range(c1):
        value=int(input('enter value : '))
        ab[r][c]=value
print(ab)
```

```
===== RESTART: C:/User:  
enter number of rows : 3  
enter number of cols : 3  
[[0 0 0]  
 [0 0 0]  
 [0 0 0]]  
>>>  
===== RESTART: C:/User:  
enter number of rows : 2  
enter number of cols : 2  
enter value : 1  
enter value : 2  
enter value : 6  
enter value : 5  
[[1 2]  
 [6 5]]
```

```
import numpy as np
r1=int(input('enter number of rows : '))
c1=int(input('enter number of cols : '))
ab=np.zeros([r1,c1],int)
mini=ab[0][0]
for r in range(r1):
    for c in range(c1):
        value=int(input('enter value : '))
        ab[r][c]=value
print(ab)
for r in range(r1):
    for c in range(c1):
        if(mini>ab[r][c]):
            mini=ab[r][c]
print('MINIMUM VALUE IS : ',mini)
```



```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 2  
AMD64) ] on win32  
Type "help", "copyright", "credits" or "  
>>>  
===== RESTART: C:/Users/IECS/Desktop/  
enter number of rows : 3  
enter number of cols : 2  
enter value : 8  
enter value : 3  
enter value : 2  
enter value : 1  
enter value : 5  
enter value : 6  
[[8 3]  
 [2 1]  
 [5 6]]  
MINIMUM VALUE IS : 0  
>>> |
```

```
import numpy as np
r1=int(input('enter number of rows : '))
c1=int(input('enter number of cols : '))
ab=np.zeros([r1,c1],int)
for r in range(r1):
    for c in range(c1):
        value=int(input('enter value : '))
        ab[r][c]=value
print(ab)
mini=ab[0][0]
for r in range(r1):
    for c in range(c1):
        if(mini>ab[r][c]):
            mini=ab[r][c]
print('MINIMUM VALUE IS : ',mini)
```

```
Python 3.8.6 (tags/v3.8.6:db4552  
AMD64) ] on win32  
Type "help", "copyright", "credi  
>>>  
===== RESTART: C:/Users/IE  
enter number of rows : 3  
enter number of cols : 2  
enter value : 7  
enter value : 6  
enter value : 5  
enter value : 1  
enter value : 3  
enter value : 4  
[[7 6]  
 [5 1]  
 [3 4]]  
MINIMUM VALUE IS : 1  
>>>
```

```
import numpy as np
ab=np.empty([3,2],dtype=int)
print(ab)
```

```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:53:45) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information
>>>
=====
RESTART: C:/Users/IECS/Desktop/p
[[ -291700632, 2046],
 [-291631392, 2046],
 [0, 0]]
```

```
import numpy as np
ab=np.empty([3,2])
print(ab)
```

```
===== RESTART: C:/Users/IECS/
[[ -291700632          2046]
 [-291631392          2046]
 [           0          0] ]
>>>
===== RESTART: C:/Users/IECS/
[[ 3.23786e-319  0.00000e+000]
 [ 0.00000e+000  0.00000e+000]
 [ 0.00000e+000  0.00000e+000] ]
>>> |
```

```
import numpy as np  
ab=np.eye(4)  
print(ab)
```

```
===== RESTART: C:/Users/IECS/Des
```

```
[ [-291700632 2046]
  [-291631392 2046]
  [ 0 0]]
```

```
>>>
```

```
===== RESTART: C:/Users/IECS/Des
```

```
[ [3.23786e-319 0.00000e+000]
  [0.00000e+000 0.00000e+000]
  [0.00000e+000 0.00000e+000]]
```

```
>>>
```

```
===== RESTART: C:/Users/IECS/Des
```

```
[ [1. 0. 0. 0.]
  [0. 1. 0. 0.] I
  [0. 0. 1. 0.]
  [0. 0. 0. 1.]]
```

```
>>>
```

```
import numpy as np  
ab=np.eye(4,dtype=int)  
print(ab)
```

```
===== RESTART: C:/Users  
[[3.23786e-319 0.00000e+000]  
 [0.00000e+000 0.00000e+000]  
 [0.00000e+000 0.00000e+000]]  
>>>  
===== RESTART: C:/Users  
[[1. 0. 0. 0.]  
 [0. 1. 0. 0.]  
 [0. 0. 1. 0.]  
 [0. 0. 0. 1.]]  
>>>  
===== RESTART: C:/Users  
[[1 0 0 0]  
 [0 1 0 0]  
 [0 0 1 0]  
 [0 0 0 1]]
```

```
import numpy as np
ab=np.eye(3, dtype=int)
print(ab)
```

```
===== RESTART:  
[[1 0 0 0]  
 [0 1 0 0]  
 [0 0 1 0]  
 [0 0 0 1]] I
```

```
>>>
```

```
===== RESTART:  
[[1 0 0]  
 [0 1 0]  
 [0 0 1]]
```

```
import numpy as np
ab=np.identity(3,dtype=int)
print(ab)
```

```
===== RESTART: C:
```

```
[ [1 0 0 0]
  [0 1 0 0]
  [0 0 1 0]
  [0 0 0 1] ]
```

```
>>>
```

```
===== RESTART: C:
```

```
[ [1 0 0]
  [0 1 0]
  [0 0 1] ]
```

```
>>>
```

```
===== RESTART: C:
```

```
[ [1 0 0]
  [0 1 0]
  [0 0 1] ]
```

```
I
```

```
import numpy as np  
ab=np.full((3, 3), 999)  
print(ab)
```

```
===== RESTART:
```

```
[[1 0 0]
 [0 1 0]
 [0 0 1]]
```

```
>>>
```

```
===== RESTART:
```

```
[[1 0 0]
 [0 1 0]
 [0 0 1]]
```

I

```
>>>
```

```
===== RESTART:
```

```
[[999 999 999]
 [999 999 999]
 [999 999 999]]
```

```
import numpy as np  
ab=np.full((3, 3), 6)  
a=5  
b=6  
print(a+b)| I
```

```
===== RESTART: C:/Users/II  
11  
>>>
```

```
import numpy as np
data=[1,7,9,8,2,3,4]
ab=np.asarray(data)
print(ab)
```

```
==> RESTART: C:/Users/IECS/AppData,  
[1 7 9 8 2 3 4]  
>>>
```

```
import numpy as np  
data=[1,9,8,2,3,4]  
ab=np.asarray(data)  
print(type(ab))
```

```
===== RESTART: C:/Users/sumit kr
<class 'numpy.ndarray'>
>>> |
```

```
>>> cd=np.array([[5, 4], [3, 6]])
>>> n=np.asmatrix(cd)
>>> cd
array([[5, 4],
       [3, 6]])
>>> n
matrix([[5, 4],
        [3, 6]])
>>> g=cd*cd
>>> g
array([[25, 16],
       [ 9, 36]])
```

```
>>> s=n*n  
>>> s  
matrix([[37, 44],  
       [33, 48]])  
>>> cd=np.array([[1,2],[3,4]])  
>>> f=np.asarray(cd)  
>>> f*f  
array([[ 1,  4],  
       [ 9, 16]])  
>>> gh=np.asmatrix(cd)  
>>> gh*gh  
matrix([[ 7, 10],  
       [15, 22]])  
>>> cd  
array([[1, 2],  
       [3, 4]])  
>>> cd[0,1]  
2  
>>> cd[1,1]  
4
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

I

Microsoft Windows [Version 6.1.7601]

Copyright (c) 2009 Microsoft Corporation. All r

C:\Users\IECS>pip3 install scipy;

```
from scipy import constants  
print(constants.kilo)
```

```
=====  RESTART: C:/  
1000.0  
>>>
```

```
from scipy import constants  
print(constants.deci)
```

```
===== RESTART: C:/User  
1000.0  
>>>  
===== RESTART: C:/User.  
0.1  
>>> t
```

```
from scipy import constants  
print(constants.centi)
```

```
===== RESTART: C:/Users/I  
1000.0  
>>>  
===== RESTART: C:/Users/I  
0.1  
>>> I  
===== RESTART: C:/Users/I  
0.01  
>>> |
```

```
for i in range(1, 6):  
    print('*|*')
```



```
Python 3.8.6 (tags/v3.8.6:db45529,  
AMD64) ] on win32  
Type "help", "copyright", "credits"  
>>>  
== RESTART: C:/Users/IECS/AppData/  
*  
*  
*  
*  
*
```

```
for i in range(1, 6):
    print('*', end=' ')
I
```

```
==> RESTART: C:/Users/IECS/1  
*****  
*****  
*****  
*****  
*****  
*****  
>>>  
==> RESTART: C:/Users/IECS/1  
*****  
*****  
*****  
*****  
*****  
>>>
```

```
for i in range(1, 6):
    print('*', end='A')
I
```

```
== RESTART: C:/Users/IECS/AppDat
```

```
*  
*  
*  
*  
*
```

```
>>>
```

```
== RESTART: C:/Users/IECS/AppDat
```

```
*****
```

```
>>>
```

```
== RESTART: C:/Users/IECS/AppDat
```

```
*A*A*A*A*A
```

```
>>> |
```

```
for i in range(1, 6):
    print('*', end=' ')
print()
for i in range(1, 6):
    print('*', end=' ')
print()
for i in range(1, 6):
    print('*', end=' ')
print()
for i in range(1, 6):
    print('*', end=' ')
print()
for i in range(1, 6):
    print('*', end=' ')
print()|
```

```
Python 3.8.6 (tags/v3.8.6:db4f1d3, Sep 24 2020, 16:46:18)
[Clang 6.0 (clang-600.0.57)] on win32
Type "help", "copyright" or "credits" for more information.

>>>
```

```
== RESTART: C:/User:
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
>>>
```

```
for j in range(1, 6):
    for i in range(1, 6):
        print('*', end=' ')
print()
```

```
==> RESTART: C:/Users/IECS/AppData
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
>>>
```

```
==> RESTART: C:/Users/IECS/AppData
```

```
*****
```

```
>>> |
```

```
for j in range(1, 6):
    for i in range(1, 6): ✓
        print('*', end=' ')
    print()
```



j
|

i
| — 5

==> RESTART: C:/Users/IECS/AppData/Local/Programs

*****~~X X K X X~~

>>> | -

4
|

i
| - 5

2
|

| - 5

1
5 ✓

```
for j in range(1, 6):
    for i in range(1, 6):
        print(j, end=' ')
    print()
```

✓
|||||
L1111

C
12345
12345



RESTART: C:/Users/IECS/AppData/Local/Programs/Python/



C
12345
12345

11111
22222
33333
44444
55555
>>>

```
for j in range(1, 6):
    for i in range(1, 6):
        print(i, end=' ')
    print()
```

12345

12345

12345

12345

12345

```
for i in range(5,0,-1):
    for j in range(5,0,-1):
        print(j,end="")
    print()
```

54321

54321

54321

54321

54321

```
for i in range(5,0,-1):
    for j in range(5,0,-1):
        print(i,end="")
    print()
```

55555

44444

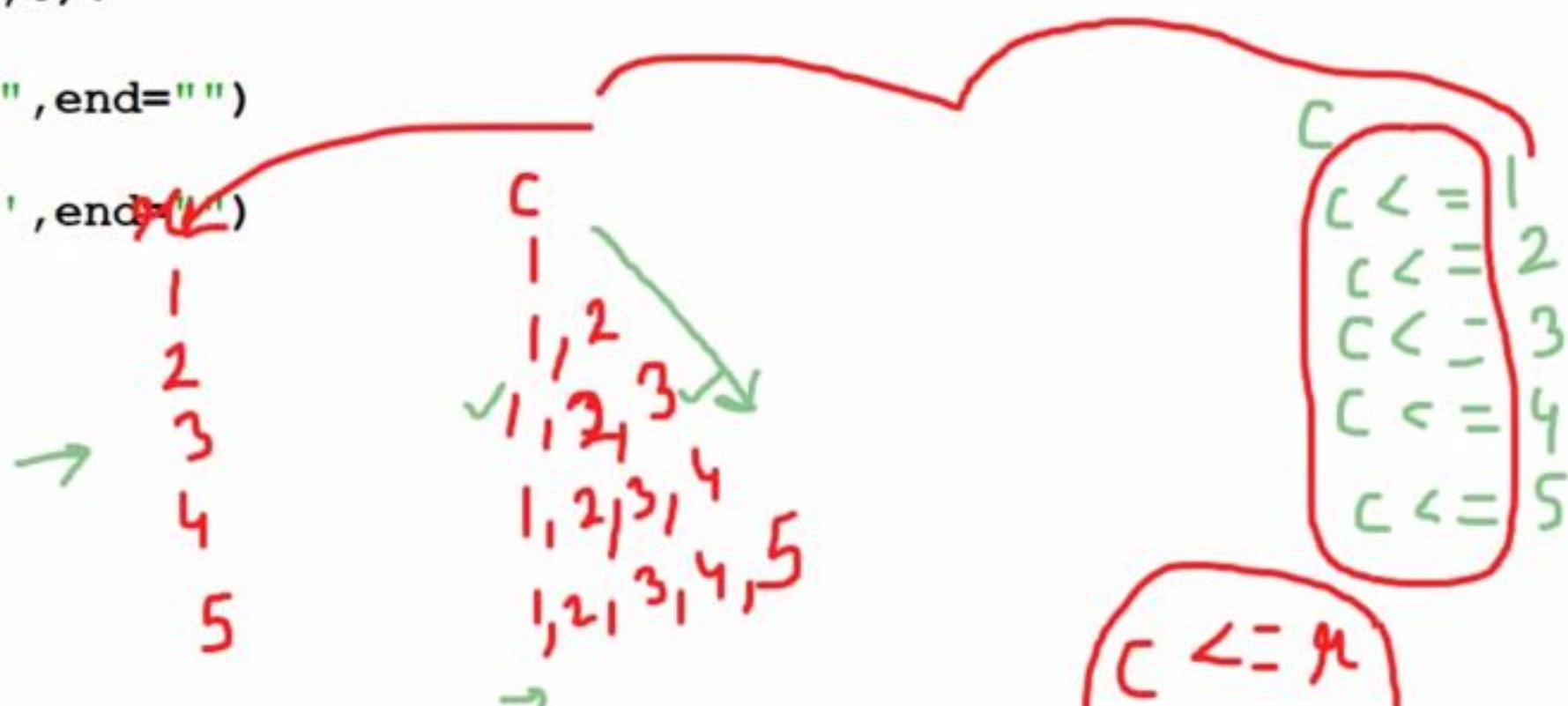
33333

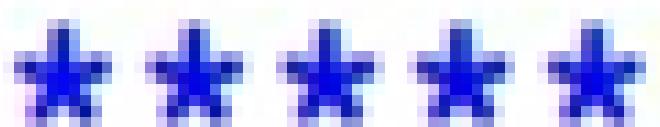
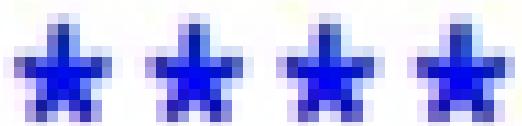
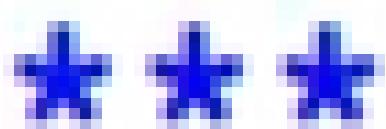
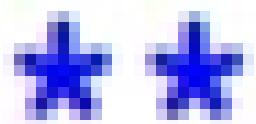
22222

11111

```
for r in range(1,6):
    for c in range(1,6):
        if(c<=r):
            print("*",end="")
        else:
            print(" ",end="")
```

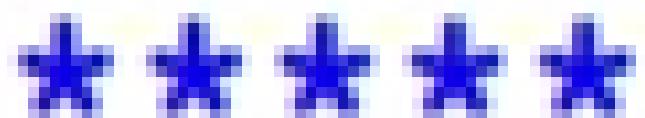
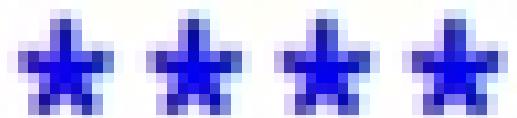
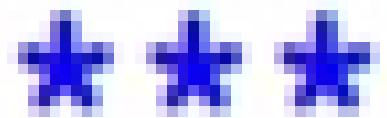
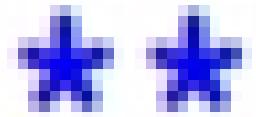
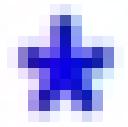
print()
1
2
3
4
5
*
**

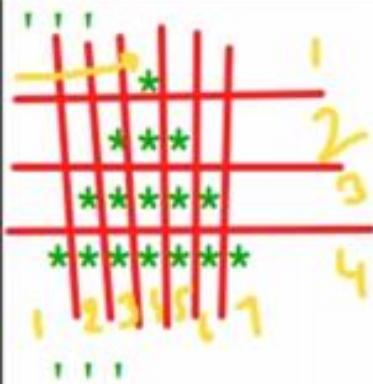




```
for r in range(1,6):
    for c in range(1,r+1):
        print("*",end="")
    print()

    for r in range(1,6):
        for c in range(1,r+1):
            if(c<=r):
                print("*",end="")
            else:
                print(' ',end="")
        print()
```

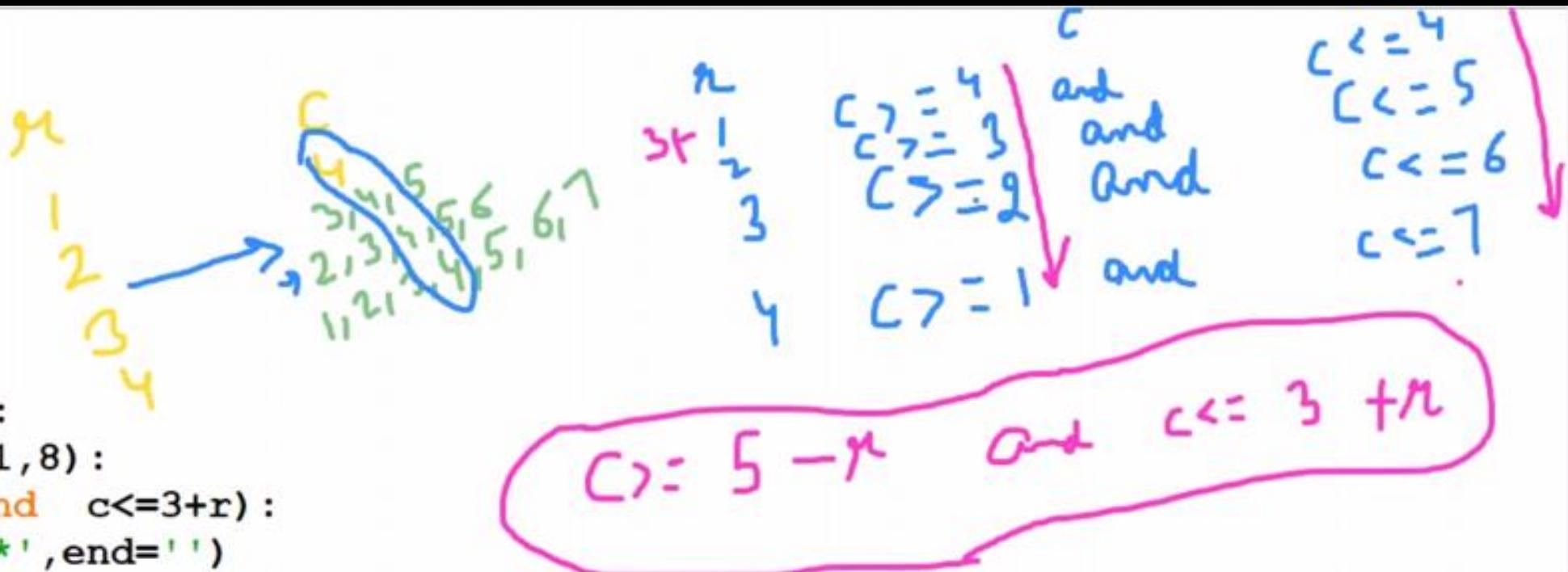


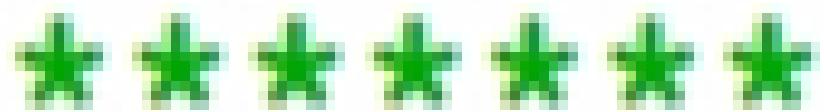
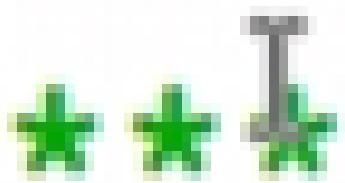
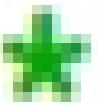


```

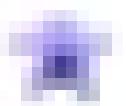
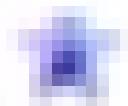
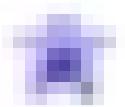
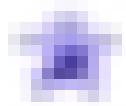
for r in range(1,5):
    for c in range(1,8):
        if(c>=5-r and c<=3+r):
            print('*',end=' ')
        else:
            print(' ',end=' ')
    print()

```

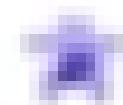
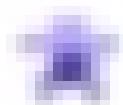
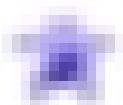
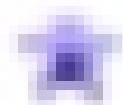
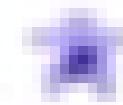
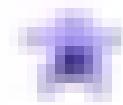
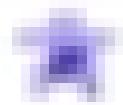
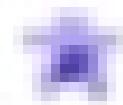
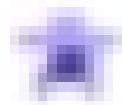
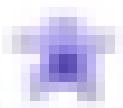




```
for r in range(1,5):
    t=1
    for c in range(1,8):
        if (c>=5-r and c<=3+r and t):
            print('**',end=' ')
            t=0
        else:
            print('  ',end=' ')
    print()
```



```
for r in range(1,5):
    t=1
    for c in range(1,8):
        if (c>=5-r and c<=3+r and t):
            print('*',end=' ')
        t=0
    else:
        print(' ',end=' ')
    t=1
print()
```



```
for i in range(1,11):
    if(i==3):
        break
    print(i)
```



1

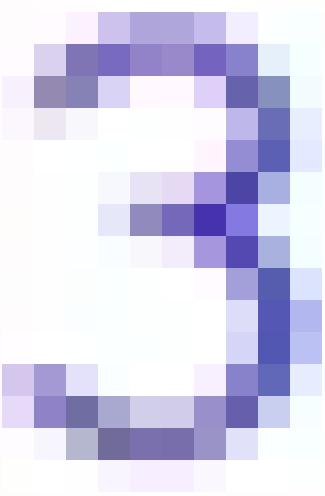
2



```
for i in range(1,11):
    if(i==3):
        continue
    print(i)
```

1
2
4
5
6
7
8
9
10

```
for i in range(1,11):
    if(i==3):
        pass
    print(i)
```



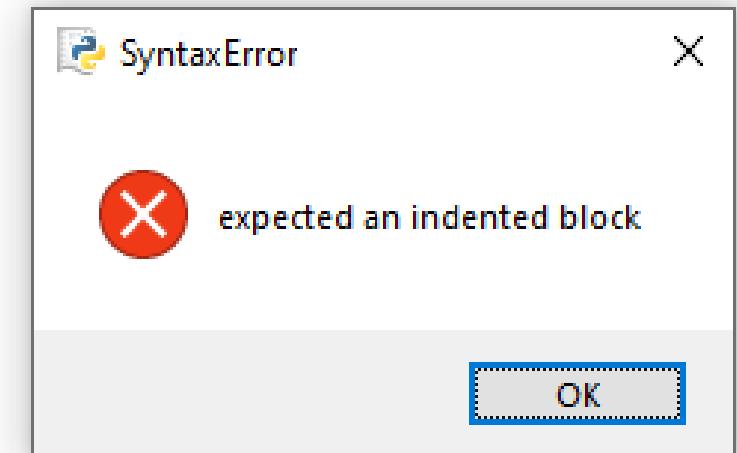
```
for i in range(1,11):
```



```
for i in range(1,11):  
    pass|
```

INTRODUCTION • CONCLUSION

```
for i in range(1,11):  
    if(i==4):  
        print(i)
```



```
for i in range(1,11):
    if(i==4):
        pass
    print(i)
```

1 2 3 4 5 6 7 8 9 10

#function :- set of code to perform a specific task.
#user defined function :-

```
def sumit():  
    print('hello')
```

```
| sumit()
```

hello

>>>

#function :- set of code to perform a specific task.
#user defined function :-

```
def sumit():
    a=int(input('enter first number : '))
    b=int(input('enter second number : '))
    c=a+b
    print('the addition of ',a,' and ',b,' is ',c)

sumit()
```

```
>>> enter first number : 66  
enter second number : 55  
the addition of 66 and 55 is 121
```

#function :- set of code to perform a specific task.
#user defined function :-

```
def sumit():
    a=int(input('enter first number : '))
    b=int(input('enter second number : '))
    c=a+b
    print('the addition of ',a,' and ',b,' is ',c)
```

```
sumit() ↴
sumit()
sumit()|
```

```
100% time. 0.77 Python 3.6.1 |Anaconda|
```

```
enter first number : 2
enter second number : 3
the addition of 2 and 3 is 5
enter first number : 4
enter second number : 5
the addition of 4 and 5 is 9
enter first number : 6
enter second number : 7
the addition of 6 and 7 is 13
```

```
#function :- set of code to perform a specific task.  
#user defined function :-
```

```
def sumit():  
    a=int(input('enter first number : '))  
    b=int(input('enter second number : '))  
    c=a+b  
    return c  
    #print('the addition of ',a,' and ',b,' is ',c)  
  
cd=sumit()  
print(cd)
```

RESUME . D .

enter first number : 44
enter second number : 33

77

>>> |

```
#function :- set of code to perform a specific task.  
#user defined function :-
```

```
def sumit():  
    a=int(input('enter first number : '))  
    b=int(input('enter second number : '))  
    c=a+b  
    return c  
#print('the addition of ',a,' and ',b,' is ',c)
```

```
cd=sumit()  
print('the addition of ',a,' and ',b,' is ',cd)
```

```
enter first number : 3
enter second number : 4
Traceback (most recent call last):
  File "D:/python batch/functiondata.py", line 12, in <module>
    print('the addition of ',a,' and ',b,' is ',cd)
NameError: name 'a' is not defined
>>> |
```

```
#function :- set of code to perform a specific task.  
#user defined function :-  
  
def sumit():  
    a=int(input('enter first number : '))  
    b=int(input('enter second number : '))  
    c=a+b  
    return 99  
    #print('the addition of ',a,' and ',b,' is ',c)  
  
cd=sumit()  
print(cd)|  
#print('the addition of ',a,' and ',b,' is ',cd)
```

```
enter first number : 33
enter second number : 44
99
```

```
def fun(a,b):  
    I #a=5  
    #b=4  
    c=a+b  
    return c  
  
res=fun(5,8)  
print(res)
```

====

RESTART: C:\Users

13

>>> |

```
def fun(a,b):  
    I #a=5  
    #b=4  
    c=a+b  
    return c  
  
a,b=4,7  
res=fun(5,8)  
print(res)
```

====

RESTART: C:\Users

13

>>> |

```
#take something , return something
def fun(a,b): #formal argument
    c=a+b
    return c
a,b=4,7           I
res=fun(a,b)      #actual argument
print(res)
```



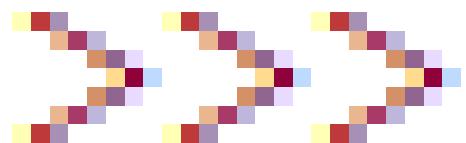
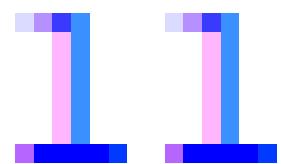
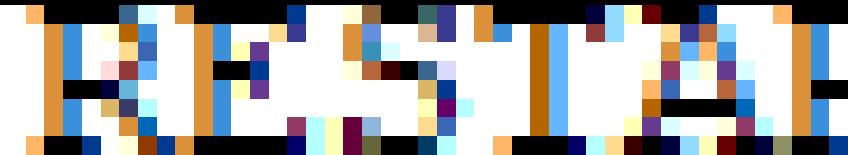
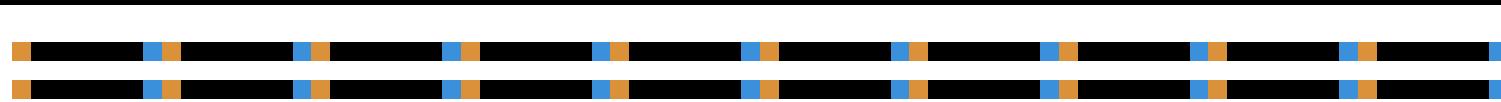
RESTART:



11



```
#take something , return nothing
def fun(a,b): #formal argument
    c=a+b
    print(c)
a,b=4,7           I
fun(a,b)         #actual argument
#print(res)
```



#function calls itself again and again

5!=5*4!

4*3!

3*2!

2*1!

```
#function calls itself again and again
120=5*24
def fact(a) : I
    if(a==0 or a==1) :
        return 1
    else:
        return a*fact(a-1)

n=int(input('enter number : '))
res=fact(n)
print(res)
```

enter number : 6
720

```
def data(age, name) :  
    print(age+5)  
    print(name)  
data(7, 'aman')
```

====

RESTART:

12

amain

====

```
def data(age, name):  
    print(age+5)  
    print(name)  
data('aman', 8)
```

```
===== RESTART: C:\Users\sumit kumar\Desktop\New folder\pattern.py =====
Traceback (most recent call last):
  File "C:\Users\sumit kumar\Desktop\New folder\pattern.py", line 4, in <module>
    data('aman',8)
  File "C:\Users\sumit kumar\Desktop\New folder\pattern.py", line 2, in data
    print(age+5)
TypeError: can only concatenate str (not "int") to str
>>> |
```

```
def data(age,name):  
    print(age+5)  
    print(name)  
data(name='aman',age=8)
```

13

amazon



```
def data(name,age=18):  
    print(age+5)  
    print(name)  
data(name='aman')  
data(name='sumit',age=21)
```

23

aman

26

sumit

```
def data(a,b):  
    a=a+b  
    print('value of a is : ',a,'and b is : ',b)
```

```
a,b=3, 4  
data(a,b)  
print('value of a is : ',a,'and b is : ',b)
```

```
--> READING: ./output/test/appData, line  
value of a is : 7 and b is : 4  
value of a is : 3 and b is : 4  
-->
```

```
>>> a=5
>>> id(a)
8791313876768
>>> a=9
>>> id(a)
8791313876896
>>> id(9)
8791313876896
```

```
def data(a,b):  
    print('id of a before calculation : ',id(a))  
    a=a+b  
  
    print('value of a is : ',a,'and b is : ',b)  
    print('id of a after calculation : ',id(a))  
  
a,b=3,4  
data(a,b)  
print('value of a is : ',a,'and b is : ',b)
```

```
id of a before calculation : 8791313876704
value of a is : 7 and b is : 4
id of a after calculation : 8791313876832
value of a is : 3 and b is : 4
>>> |
```

```
>>> id(7)
8791313876832
>>> a=None
>>> type(a)
<class 'NoneType'>
```

```
>>> ab='welcome'  
>>> type(ab)  
<class 'str'>  
>>> ab  
'welcome'  
>>> print(ab)  
welcome  
>>> ab.upper()  
'WELCOME'  
>>> cd=ab.upper()  
>>> cd  
'WELCOME'  
>>> ab  
'welcome'  
>>> ef=cd.lower()  
>>> ef  
'welcome'
```

```
>>> ab.swapcase()
'wELCOME'
>>> ef
'welcome'
>>> ef.islower()
True
>>> ef.isupper()
False
>>> ef.find('e')
1
>>> ef.find('e', 3, 6)
-1
>>> ef.find('e', 3, 7)
6
>>> ef.rfind('C')
-1
>>> ef
'welcome'
>>> ef.rfind('c')
3
```

```
>>> ef='hello hii how are you'  
>>> ef.capitalize()  
'Hello hii how are you'  
>>> ef.title()  
'Hello Hii How Are You'  
>>> ef.count('h')  
3  
>>> ef.istitle()  
False  
>>> ab=ef.title()  
>>> ab  
'Hello Hii How Are You'  
>>> ab.istitle()  
True
```

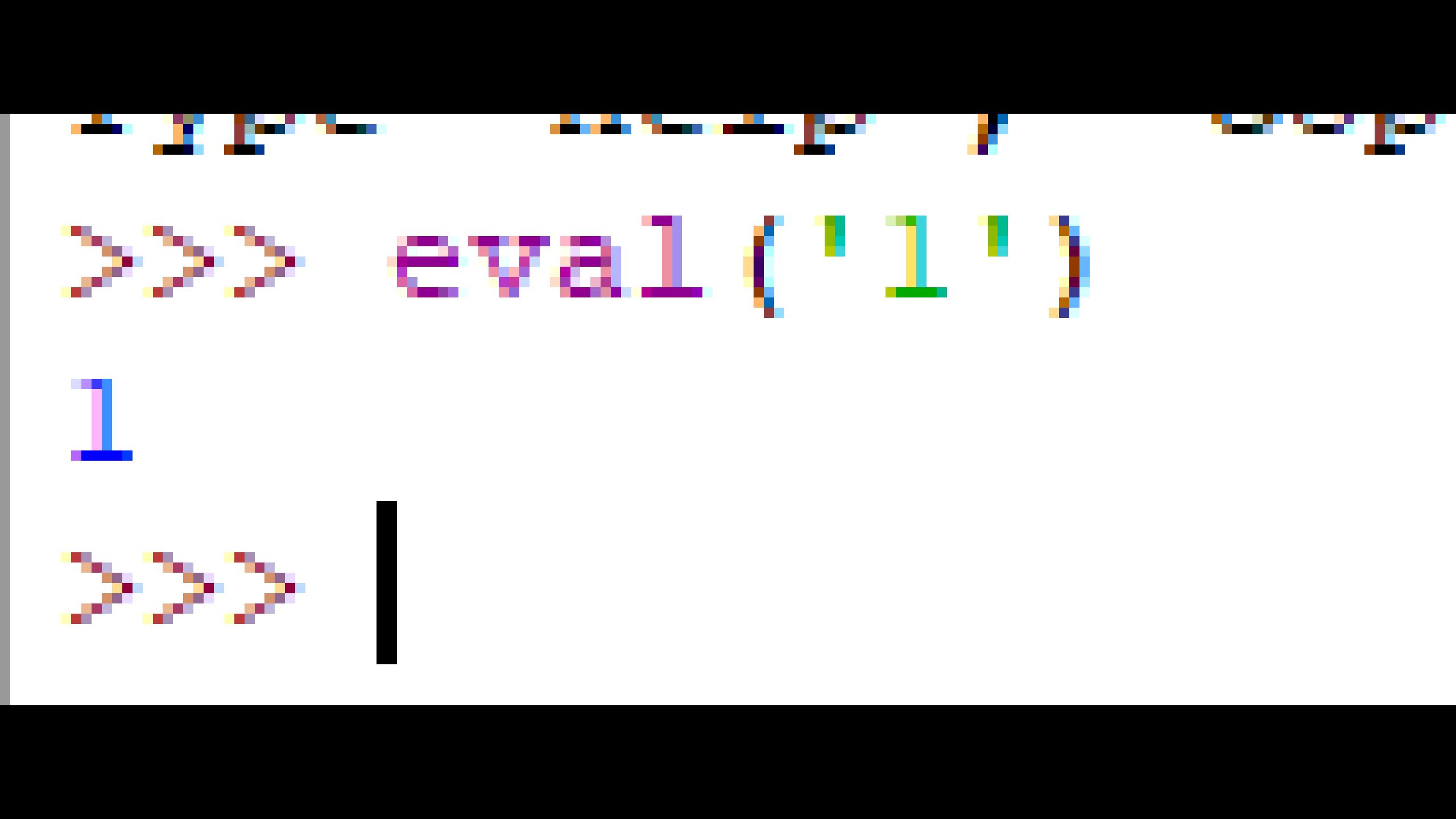
```
>>> ab
'Hello Hii How Are You'
>>> ef
'hello hii how are you'
>>> cd=ef.replace('hii','sumit')
>>> cd
'hello sumit how are you'
>>> ef.strip()
'hello hii how are you'
>>> ab="      hello      "
>>> ab.strip()
'hello'
>>> ab
'      hello      '
>>> ab.lstrip()
'hello      '
>>> ab.rstrip()
'      hello'
```

```
>>> ef
'hello hii how are you'
>>> '@'.join(ef)
'h@e@l@l@o@ @h@i@i@ @h@o@w@ @a@r@e@ @y@o@u'
>>>
```

```
>>> min(3,4)
3
>>> a=[8,7,1,2,4,5,6,7,3,2]
>>> min(a)
1
>>> max(3,4)
4
>>> max(a)
8
>>> 2**3
8
>>> pow(2,3)
8
>>> ceil(3.56)
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    ceil(3.56)
NameError: name 'ceil' is not defined
```

```
>>> import math as m
>>> m.ceil(3.56)
4
>>> m.ceil(3.01)
4
>>> m.floor(3.56)
3
>>> m.sqrt(9)
3.0
>>> import random as r
>>> r.random()
0.30184741453900055
>>> r.random()
0.7968642650331739
>>> r.random()
0.21842017583539564
>>> r.random()
0.8439342907263256
>>> r.random()
0.5429389857654576
>>>
```

```
>>> r.randrange(1,50)
16
>>> r.randrange(1,50)
35
>>> r.randrange(1,50)
29
>>> r.randrange(1,50)
47
>>> a=5.6
>>> int(a)
5
>>> int('1')
1
>>> round(3.145678)
3
>>> round(3.145678,2)
3.15
>>> round(3.145678,4)
3.1457
```



```
'''  
read -> r  
write -> w  
append ->a  
'''  
  
ab=open ('ab.txt', 'w')  
ab.write('hello')
```

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64  
D64] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:\Users\sumit kumar\Desktop\New folder\stars.py =====  
>>> |
```

```
'''  
read -> r  
write -> w  
append ->a  
'''  
  
ab=open ('ab.txt','r')  
print(ab.read())|
```

he11o

hi how are you

>>>

```
ab=open('ab.txt','r')  
print(ab.readline())  
  
print(ab.readline())  
  
print(ab.readline())
```

hello

hi how are you

hello

```
'''  
read -> r  
write -> w  
append ->a  
'''  
  
ab=open ('ab.txt','r')  
print(ab.readline(),end="")  
print(ab.readline())  
print(ab.readline())
```

hello

hi how are you

hello

!!!

read -> r

write -> w

append ->a

!!!

```
ab=open('ab.txt','r')
```

```
print(ab.readlines())
```

```
RESTART: D:/python/Batch/timedata1.py
['hello\n', 'hii how are you\n', 'hello\n', 'hii how are you\n', 'hello\n', 'hii
how are you\n', 'hello\n', 'hii how are you'] . .
>>> |
```

Activate Wi

```
'''  
read -> r  
write -> w  
append ->a
```

```
'''
```

```
ab=open('ab.txt','w')  
ab.write("byebye")  
ab.close()
```

ab - Notepad

File Edit Format View Help

byebye:



```
'''  
read -> r  
write -> w  
append ->a  
'''  
  
ab=open ('ab.txt','a')  
ab.write("sumit")|  
ab.close()
```

ab - Notepad

File Edit Format View Help

byebyesumit



File Edit Format View Help

chanchal

ritika

ashwani

subhash

sumit

ajay

anoop

|

!!!

read -> r

write -> w

append ->a

!!!

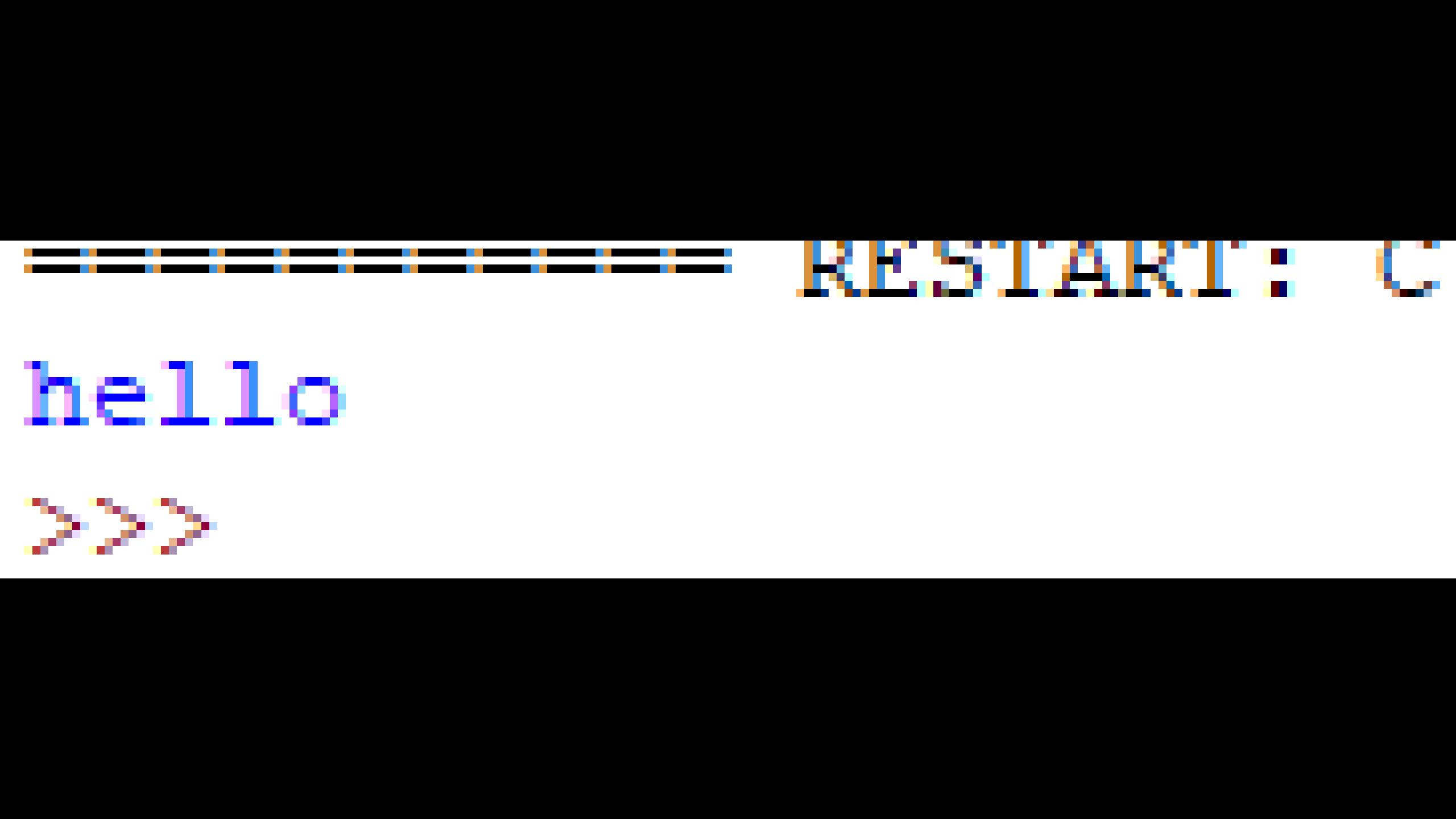
```
ab=open('ab.txt','r')
```

```
ab.seek(10)
```

```
print(ab.readline())
```

```
ab.close()
```

```
class A:  
    def abc(self):  
        print('hello')  
  
bharti=A()  
  
#bharti.abc()  
A.abc(bharti)|
```



```
class A:  
    def abc(self):  
        print('hello')  
  
bharti=A()  
bharti.abc()  
#A.abc(bharti)
```

RECOLHIMENTOS

hello

>>>

```
class A:  
    def __init__(self):  
        print('hello')
```

```
bharti=A()  
#bharti.abc()  
#A.abc(bharti)
```

RESTART

hello

>>>

RESTART

hello

>>>

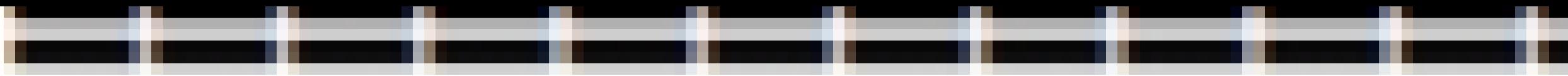
RESTART

hello

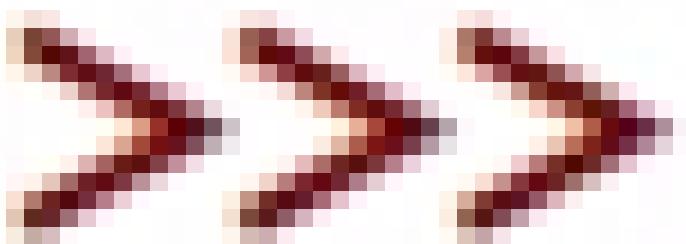
>>> |

```
class A:  
    def __init__(self,id):  
        self.id=id  
        id=555  
        print(self.id)
```

```
bharti=A(111)  
#bharti.abc()  
#A.abc(bharti)
```



1 1 1



```
#local variable  
#global variable  
def data():  
    x=55  
    print(x)
```

```
data()  
print(x)
```

55

```
Traceback (most recent call last):
  File "D:/python batch/scope.py", line 8, in <module>
    print(x)
NameError: name 'x' is not defined
>>>
```

```
#local variable  
#global variable  
x=77  
  
def data():  
    x=55  
    print(x)  
  
data()  
print(x)
```

55

77



```
'''  
scope :-  
the scope defines on which hierarchy level python  
searches for particular variable name for its associated object.
```

```
'''  
#local variable  
#global variable  
x=77  
def data():  
    x=55                                I  
    print(type(x))  
  
data()  
print(x)
```

```
<class 'int'>
```

77

>>>

LEGB->Local->Enclosed->Global->Built-in
Local can be inside a function.it is contains
names defined inside the current function.

```
'''  
#local variable  
#global variable  
x=77  
def data():  
    x=55  
    print('inside data function : ',x)  
    def fun():  
        x=67  
        print('inside fun function : ',x)  
    fun()  
  
data()  
print('outside| function : ',x)
```

inside data function : 55
inside fun function : 67
outside function : 77

```
#local variable
#globals variable
x=77
def data():
    def fun():
        x=67
        print('inside fun function : ',x)
    fun()
    x=55
    print('inside data function : ',x)

data()
print('outside function : ',x)
```

```
inside fun function : 67
inside data function : 55
outside function : 77
>>> |
```

```
#local variable
#gloabal variable
x=77
def data():
    x=55
    def fun():
        x=67
        print('inside fun function : ',x)
    fun()
    print('inside data function : ',x)
data()
```

'''

scope :-

the scope defines on which hierarchy level python searches for particular variable name for its associated object.

LEGB->Local->Enclosed->Global->Built-in

Local can be inside a function.it is contains names defined inside the current function.

Enclosed can be its enclosing function.that is-> if a function is wrapped inside function.

global refers to the uppermost level of the executing script itself.

Built-in ->are special names that are reserve for python itself.

'''

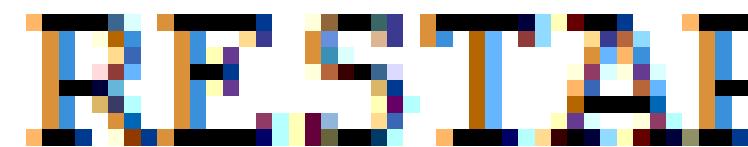
```
#local variable
#globaI variable
def len():
    return 1
x=len('ab')
print(x) I
```

1

2 3 4

```
#local variable
#globaL variable
def len(t):
    return 1
x=len('ab')
print(x)
```

 =



1





```
>>> help('keywords')
```

Here is a list of the Python keywords. Enter any keyword to get more help.

False	break	for	not
None	class	from	or
True	continue	global	pass
<u>__peg_parser__</u>	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield

scope.py - D:/python batch/scope.py (3.9.6)

File Edit Format Run Options Window Help

```
def add(a,b):
    c=a+b
    print('the addition of ',a,' and ',b,' is ',c)

add(3,4)
```

```
----- RESTART: D:/
```

```
the addition of 3 and 4 is 7
```

```
>>> |
```

The screenshot shows a code editor window with a tab labeled "sum.py - D:/python batch/sum.py (3.9.6)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The main text area contains the following code:

```
it allows you to reuse one more functions in  
your program, even in the programs in which those functions have  
not been defined.  
  
import scope as a  
a.add(4,5)
```

```
----- RESTART: D:/py  
the addition of 4 and 5 is 9  
>>>
```

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhart>cd D:

D:\

C:\Users\bhart>D:

D:\>cd D:\python batch

D:\python batch>ls

'ls' is not recognized as an internal or external command,
operable program or batch file.



D:\python batch>cd..

D:\>cd D:\python batch

D:\python batch>py commanddata.py
hello

D:\python batch>_

```
import sys  
print(sys.argv)
```

(c) Microsoft Corporation. All rights reserved.

C:\Users\bhart>cd D:

D:\

C:\Users\bhart>D:

D:\>cd D:\python batch

D:\python batch>ls

'ls' is not recognized as an internal or external command,
operable program or batch file.

D:\python batch>cd..

D:\>cd D:\python batch

D:\python batch>py commanddata.py

hello



D:\python batch>py commanddata.py 3 4 5 6 7 8 9

hello

D:\python batch>py commanddata.py 3 4 5 6 7 8 9

['commanddata.py', '3', '4', '5', '6', '7', '8', '9']

D:\python batch>py commanddata.py 3 4 5 6 7 8 9

commanddata.py

```
import sys  
a=int(sys.argv[1])  
b=int(sys.argv[2])  
print(a+b)
```

```
D:\python batch>py commanddata.py 3 4 5 6 7 8 9  
7  
D:\python batch>
```

```
>>> ab={1:'bharti',2:'Pallavi',3:'Sanjana'}
>>> type(ab)
<class 'dict'>
>>> ab[1]
'bharti'
>>> len(ab)
3
>>> ab
{1: 'bharti', 2: 'Pallavi', 3: 'Sanjana'}
>>> ab.keys()
dict_keys([1, 2, 3])
>>> ab[1]='ashwani'
>>> ab
{1: 'ashwani', 2: 'Pallavi', 3: 'Sanjana'}
>>> ab.values
<built-in method values of dict object at 0x00000176CD3CD1C0>
>>> ab.values()
dict_values(['ashwani', 'Pallavi', 'Sanjana'])
>>
```

```
>>> ab.items()
dict_items([(1, 'ashwani'), (2, 'Pallavi'), (3, 'Sanjana')])
>>> print(ab.items())
dict_items([(1, 'ashwani'), (2, 'Pallavi'), (3, 'Sanjana')])
>>> ab.get(4, 'hihi')
'hihi'
>>> ab
{1: 'ashwani', 2: 'Pallavi', 3: 'Sanjana'}
>>> ab.get(1)
'ashwani'
>>> ab.pop()
Traceback (most recent call last):
  File "<pyshell#15>", line 1, in <module>
    ab.pop()
TypeError: pop expected at least 1 argument, got 0
>>> ab.pop()2
SyntaxError: invalid syntax
>>> ab.pop(2)
'Pallavi'
```

```
>>> ab
{1: 'ashwani', 3: 'Sanjana'}
>>> ab.fromkeys()
Traceback (most recent call last):
  File "<pyshell#19>", line 1, in <module>
    ab.fromkeys()
TypeError: fromkeys expected at least 1 argument, got 0
>>> ab.fromkeys(1)
Traceback (most recent call last):
  File "<pyshell#20>", line 1, in <module>
    ab.fromkeys(1)
TypeError: 'int' object is not iterable
>>> ab.fromkeys('ashwani')
{'a': None, 's': None, 'h': None, 'w': None, 'n': None, 'i': None}
>>> h
```

```
>>> ab.fromkeys('a')
{'a': None}
>>> p= ('a', 'b', 'c', 'd')
>>> y=1
>>> cd=dict.fromkeys(p,y)
>>> cd
{'a': 1, 'b': 1, 'c': 1, 'd': 1}
```

```
for i in range(256):
    print(chr(i))
    print(value, ...,
```

í ï î ï ð ñ ò ó ô õ ö ö - ø ú ú û ü ý þ ÿ
">>>>

I

```
for i in range(256):
    print(i, '.', chr(i), end=' | ')
```

I

RESTART: D:/PyCharm/Batch/datasetcode.py

0 . 1 . ॥ 2 . 3 . ॥ 4 . ॥ 5 . ॥ 6 . ॥ 7 . ॥ 8 . ॥ 9 . 10 .
 11 . ॥ 12 . ♪ 13 . 14 . ॥ 15 . ॥ 16 . ॥ 17 . ॥ 18 . ॥ 19 . ॥ 20 . ॥ 21 . ॥ 22
 . ॥ 23 . ॥ 24 . ॥ 25 . ॥ 26 . ॥ 27 . ॥ 28 . 29 . 30 . 31 . 32 . 33 .
 ! 34 . " 35 . # 36 . \$ 37 . % 38 . & 39 . ' 40 . (41 .) 42 . * 43 . + 44 . , 4
 5 . - 46 . . 47 . / 48 . 0 49 . 1 50 . 2 51 . 3 52 . 4 53 . 5 54 . 6 55 . 7 56 .
 8 57 . 9 58 . : 59 . ; 60 . < 61 . = 62 . > 63 . ? 64 . @ 65 . A 66 . B 67 . C
 68 . D 69 . E 70 . F 71 . G 72 . H 73 . I 74 . J 75 . K 76 . L 77 . M 78 . N 79
 . O 80 . P 81 . Q 82 . R 83 . S 84 . T 85 . U 86 . V 87 . W 88 . X 89 . Y 90 . Z
 91 . [92 . \ 93 .] 94 . ^ 95 . _ 96 . ` 97 . a 98 . b 99 . c 100 . d 101 . e
 102 . f 103 . g 104 . h 105 . i 106 . j 107 . k 108 . l 109 . m 110 . n 111 . o
 112 . p 113 . q 114 . r 115 . s 116 . t 117 . u 118 . v 119 . w 120 . x 121 . y
 122 . z 123 . { 124 . | 125 . } 126 . ~ 127 . ॥ 128 . 129 . 130 . 131 . 132
 . 133 . 134 . 135 . 136 . 137 . 138 . 139 . 140 . 141 . 142 . 143 .
 144 . 145 . 146 . 147 . 148 . 149 . 150 . 151 . 152 . 153 . 154 . 155
 . 156 . 157 . 158 . 159 . 160 . 161 . ; 162 . ¢ 163 . £ 164 . ¤ 165 . ¥
 166 . ¡ 167 . § 168 . “ 169 . © 170 . ª 171 . « 172 . ¬ 173 . - 174 . ® 175 . —
 176 . ° 177 . ± 178 . ² 179 . ³ 180 . ¹ 181 . µ 182 . ¶ 183 . ˙ 184 . , 185 . ¹
 186 . ° 187 . » 188 . ¼ 189 . ½ 190 . ¾ 191 . ڏ 192 . À 193 . Á 194 . Â 195 . Ã
 196 . Ä 197 . Å 198 . Æ 199 . Ç 200 . È 201 . É 202 . Ê 203 . Ë 204 . Ì 205 . Í
 206 . Î 207 . Ï 208 . Ð 209 . Ñ 210 . Ò 211 . Ó 212 . Õ 213 . Õ 214 . Ö 215 . × Win

216 . ø 217 . Ù 218 . Ú 219 . Û 220 . Ü 221 . Ý 222 . Þ 223 . ß 224 . à 225 . á
226 . â 227 . ã 228 . ä 229 . å 230 . æ 231 . ç 232 . è 233 . é 234 . ê 235 . ë
236 . ï 237 . í 238 . î 239 . ï 240 . ð 241 . ñ 242 . ò 243 . ó 244 . ô 245 . õ
246

Activate W

```
a='A'.encode('UTF-8')  
print(a)
```



b' A'

>>



```
'''  
for i in range(256):  
    print(i,'.',chr(i),end=' ')'''  
a='A'.encode('UTF-8')  
print(a.hex())  
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

41

>>>

```
...
for i in range(256):
    print(i, '.', chr(i), end=' ')
a='A'.encode('UTF-8')
print(a.decode())
```

A

>>>

```
>>> bin(55)
```

```
'0b110111'
```

```
>>> 0b110111
```

```
55
```

```
>>>
```

```
a={ 'ab':'hello', 'gh':'hii', 'ef':'bye' }
sorted(a)
print(a)
```

```
===== RESTART: D:/python da
{'ab': 'hello', 'cd': 'hii', 'ef': 'bye'}
>>>
```

```
a={'ab':'hello','gh':'hii','ef':'bye'}  
print(sorted(a))
```

```
[ 'ab' , 'ef' , 'gh' ]
```

```
>>>
```

```
def data():
    print("hello")
```

```
data(5)
```

hello

>>>

```
def data(p):  
    print(p*p)  
  
data(5)
```

25

>>>



```
data=lambda p:print(p*p)
```

```
data(5)
```

25

>>>



```
a={'ab':'hello','cd':'hii','ef':'bye'}  
print(a['ab'])
```

hello

>>>

```
a={ 'ab':'hello', 'cd':'hi', 'ef':'bye' }
for i in a:
    print(i)
```

ab

cd

ef



```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
for i in a.values():
```

```
    print(i)
```

hello

hi i

bye

>>>

```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
for i in a.keys():
```

```
    print(i)
```

ab

cd

ef

>>>

```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
a.pop('cd')
```

```
for i in a.values():
```

```
    print(i)
```

hello

bye

>>>

```
a={'ab':'hello','cd':'hi','ef':'bye'}
```

```
a.popitem()
```

```
for i in a.values():
```

```
    print(i)
```

be11o

bii

>>>



```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
#a.popitem()
```

```
del a['ab']I
```

```
for i in a.values():
```

```
    print(i)
```

hi

bye

>>>



```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
#a.popitem()
```

```
del a
```

```
for i in a.values():
```

```
    print(i)
```

```
Traceback (most recent call last):
  File "D:/python batch/ab.py", line 4, in <module>
    for i in a.values():
NameError: name 'a' is not defined
>>>
```

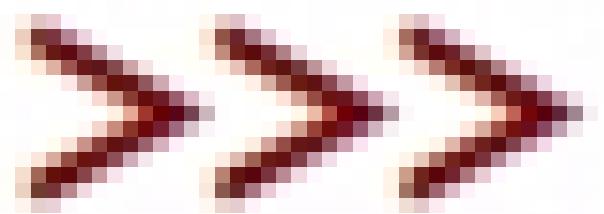
```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
#a.popitem()
```

```
a.clear()
```

```
for i in a.values():
```

```
    print(i)
```



```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
#a.popitem()
```

```
a.clear()
```

```
#for i in a.values():
```

```
print(a)
```

{ }

>>>

```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
#a.popitem()
```

```
del a
```

```
#a.clear()
```

```
#for i in a.values():
```

```
print(a)
```

```
Traceback (most recent call last):
  File "C:\Users\sumit kumar\Desktop\New folder\stars.py", line 32, in <module>
    print(a)
NameError: name 'a' is not defined
```

```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

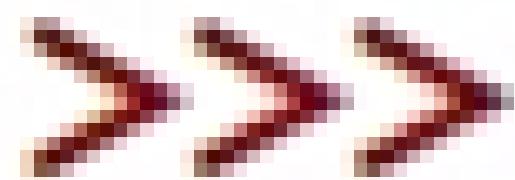
```
for i in a.values():
```

```
    print(i)
```

hello

hi i

bye



```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
a.pop('cd')
```

```
for i in a.values():
```

```
    print(i)
```

ab

cd

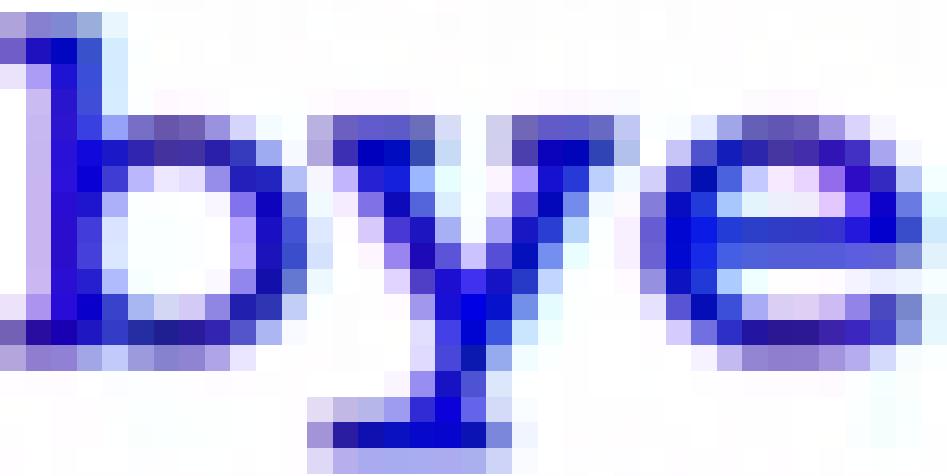
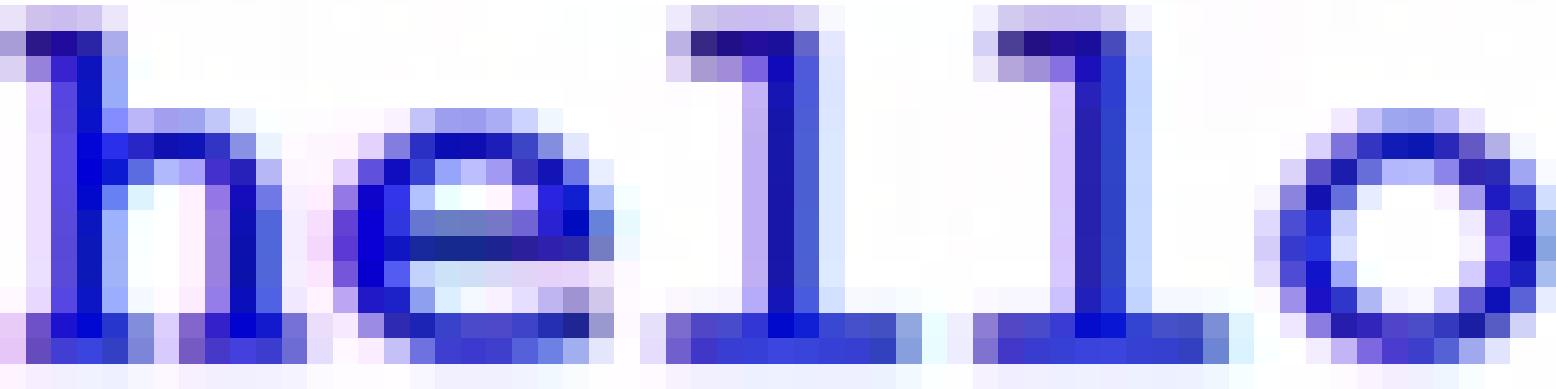
ef

```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

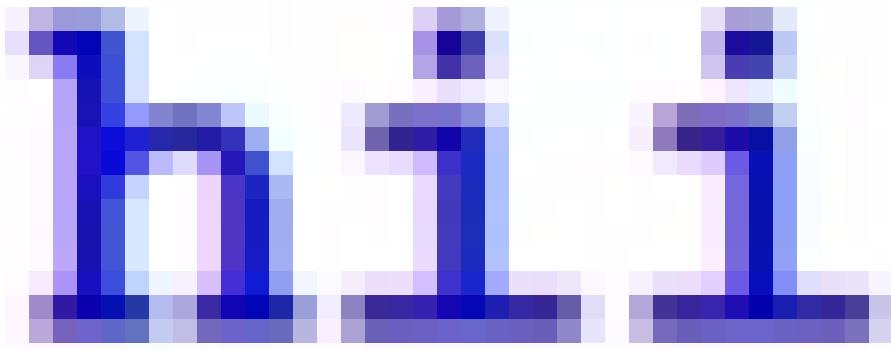
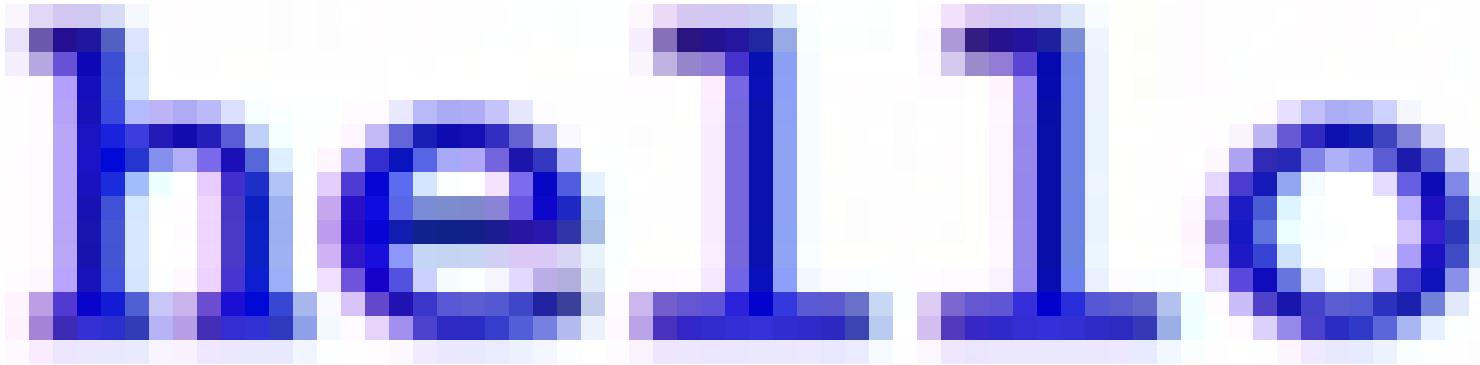
```
a.pop('cd')
```

```
for i in a.values():
```

```
    print(i)
```



```
a={'ab':'hello','cd':'hii','ef':'bye'}
a.popitem()
for i in a.values():
    print(i)
```



```
a={'ab':'hello', 'cd':'hi', 'ef':'bye' }
#a.popitem()
del a['ab']
for i in a.values():
    print(i)
```

hat

dog

```
a={'ab':'hello','cd':'hi','ef':'bye'}
```

```
#a.popitem()
```

```
del a
```

```
for i in a.values():
```

```
    print(i)
```

```
Traceback (most recent call last):
  File "D:/python batch/ab.py", line 4, in <module>
    for i in a.values():
NameError: name 'a' is not defined
>>>
```

```
a={'ab':'hello','cd':'hii','ef':'bye'}
```

```
#a.popitem()
```

```
a.clear()
```

```
for i in a.values():
```

```
    print(i)
```

>>>

```
a={'ab':'hello','cd':'hii','ef':'bye'}
#a.popitem()
a.clear()
#for i in a.values():
print(a)
```

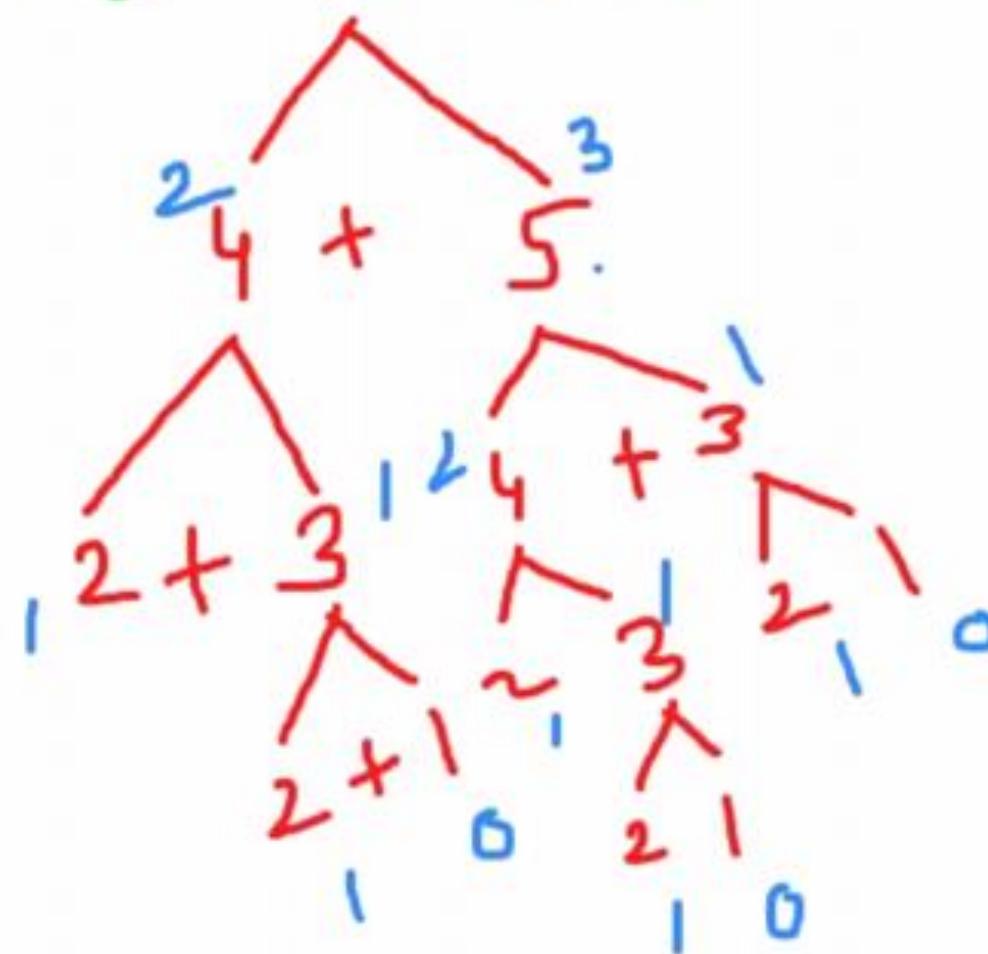
{ }

>>>

```
a={'ab':'hello','cd':'hii','ef':'bye'}
#a.popitem()
del a
#a.clear()
#for i in a.values():
print(a)
```

```
Traceback (most recent call last):
  File "C:\Users\sumit kumar\Desktop\New folder\sum.py", line 3, in <module>
    print(a)
NameError: name 'a' is not defined
>>> |
```

1 23
0 1 1 2 ✓ 3 5 8 13 21 34



```
def fib(t):
    if(t==1):
        return 0
    elif(t==2):
        return 1
    else:
        return fib(t-1)+fib(t-2)

for p in range(1,11):
    print(fib(p))
```

```
a='A'.encode('UTF-8')
print(a.hex())
print(a.decode())
```



41

A

>>>

