In [1]:
```
def greet():
    print('hello')
    print('good morning')
```

In [3]:
```
def greet():
    print('hello')
    print('good morning')
greet()
```

```
hello
good morning
```

In [5]:
```
def greet():
    print('hello')
    print('good morning')
greet()

def greet():
    print('hello')
    print('good morning')
greet()
```

```
hello
good morning
hello
good morning
```

In [7]:
```
def greet():
    print('hello')
    print('good noon')
greet()

greet()
```

```
hello
good noon
hello
good noon
```

In [9]:
```
def greet():
    print('hello')
    print('good noon')
greet()

print()

greet()
```

```
hello
good noon

hello
good noon
```

In [11]:
```
def add(x,y):
    c=x+y
    print(c)
add(5,6)
```

```
11
```

In [13]:
```
def add(x,y,z):
    c=x+y+z
    print(c)
add(1,4,5)
```

```
10
```

In [15]:
```
def greet():
    print('hello')
    print('good noon')
greet()

def add(x,y):
    c=x+y
    print(c)
add(5,4)
```

```
hello
good noon
9
```

In [17]:
```
def greet():
    print('hello')
    print('good noon')

def add(x,y):
    c=x+y
    print(c)

def sub(x,y,z):
    d = x-y-z
    print(d)

greet()
add(5,4)
sub(10,2,4)
```

```
hello
good noon
9
4
```

In [19]:
```
def greet():
    print('hello')
    print('good noon')
greet()

def add(x,y):
    c=x+y
    return c

result = add(5,4)
print(result)
```

```
hello
good noon
9
```

In [21]:
```
def greet():
    print('hello')
    print('good noon')
```

```
def add(x,y,z):
    c=x+y+z
    return c

greet()

result = add(5,4,3)
print(result)
```

```
hello
good noon
12
```

In [23]:
```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

add_sub(4,5)
```

Out[23]: (9, -1)

In [25]:
```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result = add_sub(4,5)

print(result)
print(type(result))
```

```
(9, -1)
<class 'tuple'>
```

In [27]:
```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result1,result2 = add_sub(5,4)

print(result1,result2)

print(type(result1))
print(type(result2))
```

```
9 1
<class 'int'>
<class 'int'>
```

In [29]:
```
a,b = 6,7
```

Function Arguments

In [32]:
```
def update():
    x = 8
    print(x)
update()
```

```
8
```

In [34]:
```
def update():
    x = 8
    print(x)
update(8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[34], line 4
      2     x = 8
      3     print(x)
----> 4 update(8)

TypeError: update() takes 0 positional arguments but 1 was given
```

In [36]:
```
def update(x):
    x = 8
    print(x)

update(10)
```

```
8
```

In [38]:
```
def update(x):
    x = 8
    print(x)

update(10)
```

```
8
```

In [40]:
```
def update(x):
    x = 8
    print(x)

a = 10
update(a)
print(a)
```

```
8
10
```

In [42]:
```
def update(x):
    x = 8
    #print(x)

a = 10
update(a)
print(a)
```

```
10
```

PASS BY VALUE PASS BY REFERENCE

In [45]:
```
def change(a):
    a = a+ 10
    print('inside the fun a =',a)

x = 10
print('x before calling:', x)
```

```
change(x)
print('x after calling:', x)
```

```
x before calling: 10
inside the fun a = 20
x after calling: 10
```

In [47]:
```
def change(a):
    a = a+ 10
    print('inside the fun a =',a)

a = 10
print('a before calling:', a)
change(a)
print('a after calling:', a)
```

```
a before calling: 10
inside the fun a = 20
a after calling: 10
```

In [49]:
```
def change(a):
    print('This is original a',id(a))
    a = a+ 10
    print('This is the new a =',a)
    print('inside the fun a =',a)

a = 10
print('a before calling:', a)
print('This is main a:',id(a))
change(a)
print('a after calling:', a)
```

```
a before calling: 10
This is main a: 140718252042968
This is original a 140718252042968
This is the new a = 20
inside the fun a = 20
a after calling: 10
```

In [51]:
```
def change(a):
    print('This is original a',id(a))
    a = a+ 10
    print('This is the new a =',id(a))
    print('inside the fun a =',a)

a = 10
print('a before calling:', a)
print('This is main a:',id(a))
change(a)
print('a after calling:', a)
```

```
a before calling: 10
This is main a: 140718252042968
This is original a 140718252042968
This is the new a = 140718252043288
inside the fun a = 20
a after calling: 10
```

In [53]:
```
def change(a):
    a = a+ 10
    print('This is the new a =',id(a))
    print('inside the fun a =',a)
```

```
a = 10
print('a before calling:', a)
print('This is main a:',id(a))
change(a)
print('a after calling:', a)
print('This is original a',id(a))
```

```
a before calling: 10
This is main a: 140718252042968
This is the new a = 140718252043288
inside the fun a = 20
a after calling: 10
This is original a 140718252042968
```

In [55]:
```
def change(lst):
    lst[0] = lst[0]+10
    print('inside fun =', lst)

lst = [10]
print('Before calling:', lst)
change(lst)
print('After calling:',lst)
```

```
Before calling: [10]
inside fun = [20]
After calling: [20]
```

In [57]:
```
def update(x):
    x = 8
    print('x : ', x)

a = 10
update(a)
print('a : ',a)
```

```
x :  8
a :  10
```

In [59]:
```
def update(x):
    print(id(x))
    x = 8
    #print(id(x))
    print('x', x)

a = 10
print(id(a))
update(a)
print('a',a)
```

```
140718252042968
140718252042968
x 8
a 10
```

In [61]:
```
def update(x):
    #print(id(x))
    x = 8
    print(id(x))
    print('x', x)

a = 10
```

```
print(id(a))
update(a)
print('a',a)
```

```
140718252042968
140718252042904
x 8
a 10
```

In [63]:
```
def update(x):
    x = 8

    print(id(x))
    print('x', x)

a = 10
print(id(a))

update(a)
print('a',a)
```

```
140718252042968
140718252042904
x 8
a 10
```

In [65]:
```
def update(x):
    print(id(x))
    x = 8
    print(id(x))
    print('x', x)

a = 10
print(id(a))
update(a)
print('a',a)
```

```
140718252042968
140718252042968
140718252042904
x 8
a 10
```

In [67]:
```
def update(lst):
    print(id(lst))

    lst[1] = 25
    print(id(lst))
    print('x', lst)

lst = [10,20,30]
print(id(lst))
update(lst)
print('lst',lst)
```

```
2360975757056
2360975757056
2360975757056
x [10, 25, 30]
lst [10, 25, 30]
```

In [69]:
```
def modify_integer(x):
    x = 10
    print("Inside function:", x)

my_integer = 5
modify_integer(my_integer)
print("Outside function:", my_integer)
```

```
Inside function: 10
Outside function: 5
```

In [71]:
```
def modify_integer(x):
    x = 10
    print("Inside function:", x)
    print('Inside function:',id(x))

my_integer = 5
modify_integer(my_integer)
print("Outside function:", my_integer)
print('Outside function:',id(x))
```

```
Inside function: 10
Inside function: 140718252042968
Outside function: 5
Outside function: 140718252042968
```

In [73]:
```
def modify_integer(x):
    print('original Inside function:',id(x))
    x = 10
    print("Inside function:", x)
    print('Inside function:',id(x))

my_integer = 5
modify_integer(my_integer)
print("Outside function:", my_integer)
print('Outside function:',id(x))
```

```
original Inside function: 140718252042808
Inside function: 10
Inside function: 140718252042968
Outside function: 5
Outside function: 140718252042968
```

In [75]:
```
def modify_list(my_list):
    my_list.append(4)
    print("Inside function:", my_list)

my_list = [1, 2, 3]
modify_list(my_list)
print("Outside function:", my_list)
```

```
Inside function: [1, 2, 3, 4]
Outside function: [1, 2, 3, 4]
```

In [77]:
```
def modify_list(my_list):
    print("original Inside function:", id(my_list))
    my_list.append(4)
    print("Inside function:", my_list)
    print("Inside function:", id(my_list))

my_list = [1, 2, 3]
```

```
modify_list(my_list)
print("Outside function:", my_list)
print("Outside function:", id(my_list))
```

```
original Inside function: 2360975583936
Inside function: [1, 2, 3, 4]
Inside function: 2360975583936
Outside function: [1, 2, 3, 4]
Outside function: 2360975583936
```

Types Of Arguments(ACTUAL OR FORMAL ARGUMENTS)

In [80]:
```
def add(a,b):
    c = a+b
    print(c)

add(5,6)
```

```
11
```

In [82]:
```
def add(a,b,d,e):
    c = a+b+d
    print(c)

add(5,6,7,8)
```

```
18
```

ACTUAL ARGUMENT(POSITIONAL)

In [85]:
```
def person(name,age):
    print(name)
    print(age)

person('nit', 22)
```

```
nit
22
```

In [87]:
```
def person(name,age):
    print(name)
    print(age-5)

person(22,'nit')
```

```
22
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[87], line 5
      2     print(name)
      3     print(age-5)
----> 5 person(22,'nit')

Cell In[87], line 3, in person(name, age)
      1 def person(name,age):
      2     print(name)
----> 3     print(age-5)

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

In [89]:
```
def person(name,age):
    print(name)
    print(age)

person(20,'nit')
```

```
20
nit
```

def person(name,age): print(name) print(age-5) person('nit',20)

ACTUAL(kEYWORD)

In [94]:
```
def person(name,age):
    print(name)
    print(age-5)

person(20,'nit')
```

```
20
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[94], line 5
      2     print(name)
      3     print(age-5)
----> 5 person(20,'nit')

Cell In[94], line 3, in person(name, age)
      1 def person(name,age):
      2     print(name)
----> 3     print(age-5)

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

In [96]:
```
def person(name,age):
    print(name)
    print(age-5)

person(age = 20, name = 'nit')
```

```
nit
15
```

In [98]:
```
def person(name, age):
    print(name)
    print(age)

person(age = 20,  name = 'nit')
```

```
nit
20
```

In [100…
```
def person(name, age, age2):
    print(name)
    print(age)
    print(age2)

person(age = 20,  name = 'nit', age2 = 10)
```

```
nit
20
10
```

In [102…
```
x = 27
y = 4
print(x % y * 3)
```

```
9
```

Actual(DEFAULT)

In [105…
```
def person(name,age):
    print(name)
    print(age)

person('nit')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[105], line 5
      2     print(name)
      3     print(age)
----> 5 person('nit')

TypeError: person() missing 1 required positional argument: 'age'
```

In [107…
```
def person(name,age=18):
    print(name)
    print(age)

person('nit',20)
```

```
nit
20
```

In [109…
```
def person(name,age = 18):
    print(name)
    print(age)

person('nit',24)
```

```
nit
24
```

Actual(VARIABLE LENGTH)

In [112…
```
def sum(a, b):
    c = a+b
    print(c)

sum(5,6)
```

```
11
```

In [114…
```
def sum(a, b):
    c = a+b
    print(c)

sum(5,6,7,8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[114], line 5
      2     c = a+b
      3     print(c)
----> 5 sum(5,6,7,8)

TypeError: sum() takes 2 positional arguments but 4 were given
```

In [116…

```
def sum(a, *b):
    c = a+b
    print(c)

sum(5,6,7,8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[116], line 5
      2     c = a+b
      3     print(c)
----> 5 sum(5,6,7,8)

Cell In[116], line 2, in sum(a, *b)
      1 def sum(a, *b):
----> 2     c = a+b
      3     print(c)

TypeError: unsupported operand type(s) for +: 'int' and 'tuple'
```

In [118…

```
def sum(a, *b):
    #c = a+b
    print(type(a))
    print(type(b))

sum(5,6,7,8)
```

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

In [120…

```
def sum(a, *b):
    #c = a+b
    print(a)
    print(b)

sum(5,6,7,8)
```

```
5
(6, 7, 8)
```

In [122…

```
def sum(a, *b):
    c = a
    for i in b:
        c = c + i
    print(c)

sum(5,6,7,8)
```

```
26
```

In [124…

```
def sum(a, *b):

```

```
    c = a
    for i in b:
        c = c + i
    print(c)

sum(5,6,7,8,9)
```

```
35
```

In [126…

```
def sum(a, *b):

    c = 0

    for i in b:
        c = c + i
    print(c)

sum(5,6,7,8)
```

```
21
```

In [128…

```
def sum(a, *b):

    c = 0
    for i in b:
        c = c + i
    print(c)

sum(5,6,7,8,5,7,7,89,-100,23)
```

```
52
```

KWARGS

In [131…

```
def person():
    person('ALEX', 36, 'JOHN', 987767)
```

In [133…

```
def person(name,*data):
    print(name)
    print(data)

person('ALEX', 36, 'JOHN', 987767)
```

```
ALEX
(36, 'JOHN', 987767)
```

In [137…

```
def person(name,*data):
    print('name')
    print(data)
person('ALEX', age = 36, home_place ='southcity', mob =987767)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[137], line 4
      2     print('name')
      3     print(data)
----> 4 person('ALEX', age = 36, home_place ='southcity', mob =987767)

TypeError: person() got an unexpected keyword argument 'age'
```

In [139…
```
def person(name, **data):
    print(name)
    print(data)

person('mark', age = 36, home_place ='southcity', mob =987767)
```

```
mark
{'age': 36, 'home_place': 'southcity', 'mob': 987767}
```

In [141…
```
def person(name,**data):
    print('name')
    print(data)

person('mark', age = 36, home_place ='southcity', mob =987767, edu='phd', actor
```

```
name
{'age': 36, 'home_place': 'southcity', 'mob': 987767, 'edu': 'phd', 'actor': 'joh
n'}
```

In [143…
```
def person(name, **data):
    print(name)

    for i, j in data.items():
        print(i, j)

person('john', age = 36, home_place ='southcity', mob =987767, place = 'USA')
```

```
john
age 36
home_place southcity
mob 987767
place USA
```

GLOBAL OR LOCAL VARIABLE

In [146…
```
a = 10

print(a)
```

```
10
```

In [150…
```
a = 10  #-- globla variable

def something():
    b = 15 #local variable
    print('in function',b)
    print('out function',a)
```

In [152…
```
a = 10

def something():
    b = 15
    print('in function',b)

print('out function',a)
```

```
out function 10
```

In [154…
```
a = 10

def something():
```

```
    a = 15

print('in function',a)

print('out function',a)
```

```
in function 10
out function 10
```

In [156…

```
a = 10

def something():
    a = 15
    b = 8
    print(b)
    print(a)
```

In [158…

```
a = 10

def something():
    a = 15
    print(a)

print(a)
```

```
10
```

In [160…

```
a = 10

def something():
    a = 15
    print('in function',a)

print('out function',a)
```

```
out function 10
```

In [162…

```
a = 10

def something():
    b = 15
    print('in function',b)

something()

print('out function',a)
```

```
in function 15
out function 10
```

In [164…

```
a = 10

def something():
    print('in function',a)

something()
print('out function',a)
```

```
in function 10
out function 10
```

In [166…
```
a = 10

def something():
    b= 25
    print('in function',b)

something()
print('out function',a)
```

```
in function 25
out function 10
```

In [168…
```
a = 10

def something():
    a = 55
    print('in function',a)
something()

print('out function',a)
```

```
in function 55
out function 10
```

In [172…
```
a = 10

def something():
    global a
    b = 15
    print('in function',b)
    print('global variable', a)

something()

print('out function',a)
```

```
in function 15
global variable 10
out function 10
```

In [178…
```
a = 20

def something():
    global a
    a = 15
    print('in function',a)

    a = 9
something()
print('out function',a)
```

```
in function 15
out function 9
```

In [180…
```
import keyword
keyword.kwlist
```

Out[180…
```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [182…
```
a = 10
print(id(a))

def something():
    a = 9
    x = globals()['a']

    print(id(x))
    print('in function',a)


something()
print('out function',a)
```

```
140718252042968
140718252042968
in function 9
out function 10
```

In [184…
```
a = 10
print(id(a))

def something():
    a = 9
```

```
    x = globals()

    print(id(x))
    print('in function',a)

    globals()['a'] = 15


something()
print('out function',a)
```

```
140718252042968
2360920724096
in function 9
out function 15
```

Global Variable

In [187…

```
x = 10

def update_x():
    global x
    x += 5

update_x()
print(x)
```

```
15
```

In [189…

```
x = 10

def update_x():
    globals()['x'] += 5

update_x()
print(x)
```

```
15
```

Pass List TO Function

In [192…

```
def count(lst):

    even = 0
    odd = 0

    for i in lst:
        if i%2 == 0:
            even += 1
        else:
            odd +=1
    return even,odd

lst = [1, 2, 3, 4, 5 ]
even, odd = count(lst)

print(even)
print(odd)
```

```
2
3
```

In [194…
```
def count(lst):

    even = 0
    odd = 0

    for i in lst:
        if i%2 == 0:
            even += 1
        else:
            odd +=1
    return even,odd

lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13]
even,odd = count(lst)

print("Even Number: {} and odd Number : {}".format(even,odd))
```

```
Even Number: 6 and odd Number : 7
```

Fibonacci Sequence

In [197…
```
def fib(n):
    print(0)
    print(1)

fib(0)
```

```
0
1
```

In [199…
```
def fib(n):
    print(0)
    print(1)
    print(1)
    print(2)
    print(3)
    print(5)

fib(0)
```

```
0
1
1
2
3
5
```

In [201…
```
def fib(n):
    a = 0
    b = 1

    print(a)
    print(b)

    for i in range(0, n):
        c = a + b
        a = b
        b = c

        print(c)
```

```
fib(5)
```

```
0
1
1
2
3
5
8
```

In [205…
```
def fib(n):
    a, b = 0, 1
    if n == 1:
        print(a)
    else:
        print(a)
        print(b)

        for i in range(2, n):
            c = a + b
            a = b
            b = c
            print(c)
fib(2)
```

```
0
1
```

Factorial Of Number In Python

In [208…
```
def fact(n):
    f = 1
    for i in range(1, n+1):
        f = f*i

    return f

x = 4
result = fact(x)
print(result)
```

```
24
```

Recurrsion Function Calling Itself

In [211…
```
def wish():
    print('hello')
    wish()

wish()
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[211], line 5
      2     print('hello')
      3     wish()
----> 5 wish()

Cell In[211], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[211], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

    [... skipping similar frames: wish at line 3 (2972 times)]

Cell In[211], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[211], line 2, in wish()
      1 def wish():
----> 2     print('hello')
      3     wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [213…

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

wish()

wish()
```

```
hello
hello
```

In [215…

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

wish()
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[215], line 5
      2     print('hello')
      3     wish()
----> 5 wish()

Cell In[215], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[215], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

    [... skipping similar frames: wish at line 3 (2972 times)]

Cell In[215], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[215], line 2, in wish()
      1 def wish():
----> 2     print('hello')
      3     wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [217…
```
import sys
sys.getrecursionlimit()
```

Out[217…
```
3000
```

In [219…
```
import sys
sys.setrecursionlimit(200)
print(sys.getrecursionlimit())
```

```
200
```

In [221…
```
import sys
sys.getrecursionlimit()
```

Out[221…
```
200
```

In [223…
```
def wish():
    print('hello')
    wish()
wish()
```

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[223], line 4
      2     print('hello')
      3     wish()
----> 4 wish()

Cell In[223], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[223], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

    [... skipping similar frames: wish at line 3 (172 times)]

Cell In[223], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[223], line 2, in wish()
      1 def wish():
----> 2     print('hello')
      3     wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [225…

```python
import sys
sys.setrecursionlimit(150)
print(sys.getrecursionlimit())

i = 0

def wish():
    global  i
    i += 1
    print('hello', i)
    wish()
wish()
```

```
150
hello 1
hello 2
hello 3
hello 4
hello 5
hello 6
hello 7
hello 8
hello 9
hello 10
hello 11
hello 12
hello 13
hello 14
hello 15
hello 16
hello 17
hello 18
hello 19
hello 20
hello 21
hello 22
hello 23
hello 24
hello 25
hello 26
hello 27
hello 28
hello 29
hello 30
hello 31
hello 32
hello 33
hello 34
hello 35
hello 36
hello 37
hello 38
hello 39
hello 40
hello 41
hello 42
hello 43
hello 44
hello 45
hello 46
hello 47
hello 48
hello 49
hello 50
hello 51
hello 52
hello 53
hello 54
hello 55
hello 56
hello 57
hello 58
hello 59
```

hello 60
hello 61
hello 62
hello 63
hello 64
hello 65
hello 66
hello 67
hello 68
hello 69
hello 70
hello 71
hello 72
hello 73
hello 74
hello 75
hello 76
hello 77
hello 78
hello 79
hello 80
hello 81
hello 82
hello 83
hello 84
hello 85
hello 86
hello 87
hello 88
hello 89
hello 90
hello 91
hello 92
hello 93
hello 94
hello 95
hello 96
hello 97
hello 98
hello 99
hello 100
hello 101
hello 102
hello 103
hello 104
hello 105
hello 106
hello 107
hello 108
hello 109
hello 110
hello 111
hello 112
hello 113
hello 114
hello 115
hello 116
hello 117
hello 118
hello 119

```
hello 120
hello 121
hello 122
hello 123
hello 124
hello 125
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[225], line 12
     10     print('hello', i)
     11     wish()
---> 12 wish()

Cell In[225], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[225], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

    [... skipping similar frames: wish at line 11 (122 times)]

Cell In[225], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[225], line 10, in wish()
      8 global  i
      9 i += 1
---> 10 print('hello', i)
     11 wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [227…

```python
i = 0

def wish():
    global  i
    i += 1
    print('hello', i)
    wish()
wish()
```

hello 1
hello 2
hello 3
hello 4
hello 5
hello 6
hello 7
hello 8
hello 9
hello 10
hello 11
hello 12
hello 13
hello 14
hello 15
hello 16
hello 17
hello 18
hello 19
hello 20
hello 21
hello 22
hello 23
hello 24
hello 25
hello 26
hello 27
hello 28
hello 29
hello 30
hello 31
hello 32
hello 33
hello 34
hello 35
hello 36
hello 37
hello 38
hello 39
hello 40
hello 41
hello 42
hello 43
hello 44
hello 45
hello 46
hello 47
hello 48
hello 49
hello 50
hello 51
hello 52
hello 53
hello 54
hello 55
hello 56
hello 57
hello 58
hello 59
hello 60

hello 61
hello 62
hello 63
hello 64
hello 65
hello 66
hello 67
hello 68
hello 69
hello 70
hello 71
hello 72
hello 73
hello 74
hello 75
hello 76
hello 77
hello 78
hello 79
hello 80
hello 81
hello 82
hello 83
hello 84
hello 85
hello 86
hello 87
hello 88
hello 89
hello 90
hello 91
hello 92
hello 93
hello 94
hello 95
hello 96
hello 97
hello 98
hello 99
hello 100
hello 101
hello 102
hello 103
hello 104
hello 105
hello 106
hello 107
hello 108
hello 109
hello 110
hello 111
hello 112
hello 113
hello 114
hello 115
hello 116
hello 117
hello 118
hello 119
hello 120

```
hello 121
hello 122
hello 123
hello 124
hello 125
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[227], line 8
      6     print('hello', i)
      7     wish()
----> 8 wish()

Cell In[227], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

Cell In[227], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

    [... skipping similar frames: wish at line 7 (122 times)]

Cell In[227], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

Cell In[227], line 6, in wish()
      4 global  i
      5 i += 1
----> 6 print('hello', i)
      7 wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [229…
```
import sys
sys.getrecursionlimit()
```

Out[229…
```
150
```

In [231…
```
def wish():
    print('hello')
    wish()
wish()
```

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

```
hello
hello
hello
hello
hello
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[231], line 4
      2     print('hello')
      3     wish()
----> 4 wish()

Cell In[231], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[231], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

    [... skipping similar frames: wish at line 3 (122 times)]

Cell In[231], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[231], line 2, in wish()
      1 def wish():
----> 2     print('hello')
      3     wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [233…
```
import sys
print(sys.getrecursionlimit())
```

```
150
```

In [235…
```
sys.setrecursionlimit(1000)
```

In [237…
```
print(sys.getrecursionlimit())
```

```
1000
```

In [241…
```
import sys
sys.setrecursionlimit(150)
print(sys.getrecursionlimit())

i = 0
```

```
def wish():
    global  i
    i += 1
    print('hello', i)
    wish()
wish()
```

```
150
hello 1
hello 2
hello 3
hello 4
hello 5
hello 6
hello 7
hello 8
hello 9
hello 10
hello 11
hello 12
hello 13
hello 14
hello 15
hello 16
hello 17
hello 18
hello 19
hello 20
hello 21
hello 22
hello 23
hello 24
hello 25
hello 26
hello 27
hello 28
hello 29
hello 30
hello 31
hello 32
hello 33
hello 34
hello 35
hello 36
hello 37
hello 38
hello 39
hello 40
hello 41
hello 42
hello 43
hello 44
hello 45
hello 46
hello 47
hello 48
hello 49
hello 50
hello 51
hello 52
hello 53
hello 54
hello 55
hello 56
hello 57
hello 58
hello 59
```

hello 60
hello 61
hello 62
hello 63
hello 64
hello 65
hello 66
hello 67
hello 68
hello 69
hello 70
hello 71
hello 72
hello 73
hello 74
hello 75
hello 76
hello 77
hello 78
hello 79
hello 80
hello 81
hello 82
hello 83
hello 84
hello 85
hello 86
hello 87
hello 88
hello 89
hello 90
hello 91
hello 92
hello 93
hello 94
hello 95
hello 96
hello 97
hello 98
hello 99
hello 100
hello 101
hello 102
hello 103
hello 104
hello 105
hello 106
hello 107
hello 108
hello 109
hello 110
hello 111
hello 112
hello 113
hello 114
hello 115
hello 116
hello 117
hello 118
hello 119

```
hello 120
hello 121
hello 122
hello 123
hello 124
hello 125
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[241], line 12
     10     print('hello', i)
     11     wish()
---> 12 wish()

Cell In[241], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[241], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

    [... skipping similar frames: wish at line 11 (122 times)]

Cell In[241], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[241], line 10, in wish()
      8 global  i
      9 i += 1
---> 10 print('hello', i)
     11 wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

Factorial Using Recursion

In [244…

```
def fact(n):
    if n==0:
        return 1
    return n * fact(n-1)

result = fact(4)

result
```

Out[244… 24

ANONYMOUS FUNCTION OR LAMBDA

In [247…
```
def square(a):
    return a * a

result = square(5)
print(result)
```

```
25
```

In [249…
```
f = lambda a : a * a
result = f(5)
result
```

Out[249… 25

In [251…
```
f = lambda a, b : a + b
result = f(1,4)
print(result)
```

```
5
```

In [253…
```
f = lambda a, b : a + b
f1 = lambda a, b : a - b

result = f(1,4)
result1 = f1(2, 3)

print(result)
print(result1)
```

```
5
-1
```

In [255…
```
import keyword
keyword.kwlist
```

Out[255…

```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

Lambda Functions(FILTER(), MAP(), REDUCE())

In [259…

```
nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[259], line 3
      1 nums = [3,2,6,8,4,6,2,9]
----> 3 evens = list(filter(is_even, nums))

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

In [261…

```
def is_even(n):
    return n % 2 == 0

nums = [3,2,6,8,4,6,2,9]
evens = list(filter(is_even, nums))

print(evens)
```

```
[2, 6, 8, 4, 6, 2]
```

In [263…
```
def is_odd(n):
    return n % 2 != 0

nums = [3,2,6,8,4,6,2,9]

odd = list(filter(is_odd, nums))
print(odd)
```

```
[3, 9]
```

In [265…
```
nums = [3,2,6,8,4,6,2,9]
evens = list(filter(lambda n : n%2 == 0, nums))
print(evens)
```

```
[2, 6, 8, 4, 6, 2]
```

In [267…
```
nums = [3,2,6,8,4,6,2,9]
odd = list(filter(lambda n : n%2 !=0, nums))
print(odd)
```

```
[3, 9]
```

In [269…
```
nums = [3,2,6,8,4,6,2,9]

evens = list(filter(lambda n : n%2 ==0, nums))
odd = list(filter(lambda n : n%2 !=0, nums))

print(evens)
print(odd)
```

```
[2, 6, 8, 4, 6, 2]
[3, 9]
```

In [271…
```
def update(n):
    return n+2

nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))
double = list(map(update, evens))

print(evens)
print(double)
```

```
[2, 6, 8, 4, 6, 2]
[4, 8, 10, 6, 8, 4]
```

In [273…
```
nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))
double = list(map(lambda n : n*2, evens))

print(evens)
print(double)
```

```
[2, 6, 8, 4, 6, 2]
[4, 12, 16, 8, 12, 4]
```

In [275…
```
nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))
double = list(map(lambda n : n*2, evens))
```

```
double_ = list(map(lambda n : n+2, evens))
doubble_1 = list(map(lambda n : n-2, evens))

print(evens)
print(double)
print(double_)
print(doubble_1)
```

```
[2, 6, 8, 4, 6, 2]
[4, 12, 16, 8, 12, 4]
[4, 8, 10, 6, 8, 4]
[0, 4, 6, 2, 4, 0]
```

In [277…

```
nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))

double = list(map(lambda n : n*2, evens))
double_ = list(map(lambda n : n-2, evens))

print(double)
print(double_)
```

```
[4, 12, 16, 8, 12, 4]
[0, 4, 6, 2, 4, 0]
```

In [279…

```
nums = [3,2,6,8,4,6,2,9]
evens = list(filter(is_even, nums))

double = list(map(lambda n : n*2, evens))
double_ = list(map(lambda n : n-2, evens))
double1 = list(map(lambda n : n+2, evens))

print(double)
print(double_)
print(double1)
```

```
[4, 12, 16, 8, 12, 4]
[0, 4, 6, 2, 4, 0]
[4, 8, 10, 6, 8, 4]
```

In [281…

```
from functools import reduce

def add_all(a,b):
    return a+b

nums = [3,2,6,8,4,6,2]

evens = list(filter(is_even, nums))
double = list(map(lambda n : n+2, evens))

sums = reduce(add_all, double)
sums
print(sums)
```

```
40
```

In [283…

```
from functools import reduce

nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))
```

```
double = list(map(lambda n : n*2, evens))
sums = (reduce(lambda a,b : a + b, double))

print(evens)
print(double)
print(sums)
```

```
[2, 6, 8, 4, 6, 2]
[4, 12, 16, 8, 12, 4]
56
```

Python Decorators

In [286…

```
def div(a,b):
    print(a / b)
div(4,2)
```

```
2.0
```

In [288…

```
def div(a, b):
    print(a / b)
div(2,4)
```

```
0.5
```

In [290…

```
def div(a,b):

    if a<b:
        a,b = b,a
    print(a / b)

div(4,2)
```

```
2.0
```

In [292…

```
def div(a,b):
    print(a / b)

def div_decorator(func):
    def inner(a,b):
        if a<b:
            a,b = b,a
        return func(a,b)
    return inner

div = div_decorator(div)

div(2,4)
```

```
2.0
```

In [295…

```
def my_decorator(func):
    def wrapper():
        print("Something is happening before the function is called.")
        #func()
        print("Something is happening after the function is called.")
    return wrapper

@my_decorator
def say_hello():
    print("Hello!")
```

```
say_hello()
```

```
Something is happening before the function is called.
Something is happening after the function is called.
```

In [297…
```
def my_decorator(func):
    def wrapper():
        print("Something is happening before the function is called.")
        func()
        print("Something is happening after the function is called.")
    return wrapper

@my_decorator
def say_hello():
    print("Hello!")

say_hello()
```

```
Something is happening before the function is called.
Hello!
Something is happening after the function is called.
```

Modules

In [300…
```
__name__
```

Out[300…
```
'__main__'
```

In [302…
```
print(__name__)
```

```
__main__
```

In [ ]:
```
```