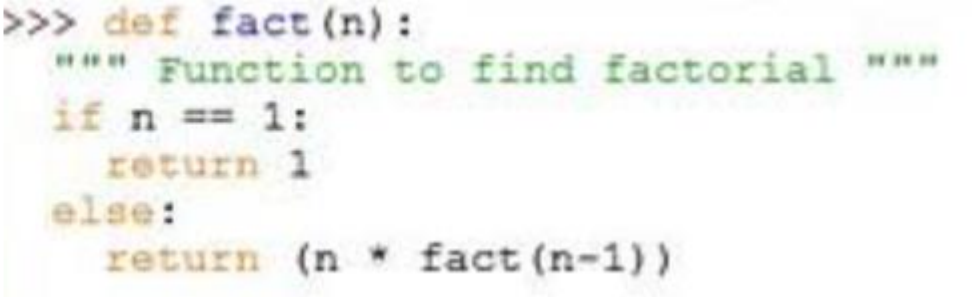
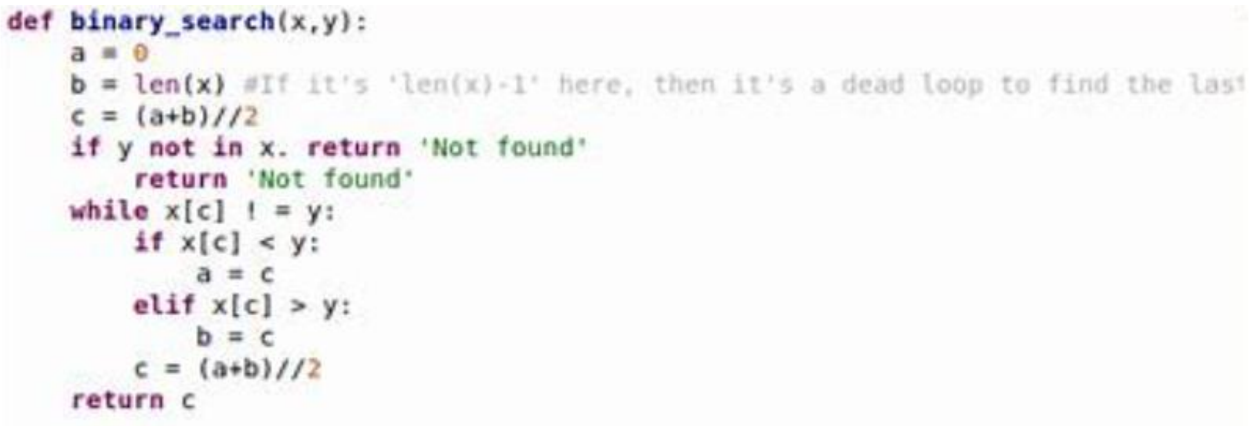
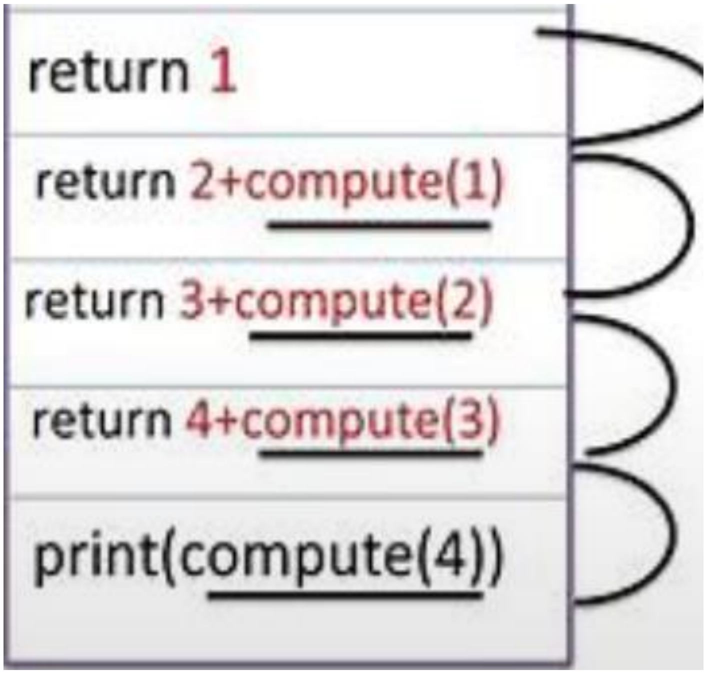
Python Recursion Examples



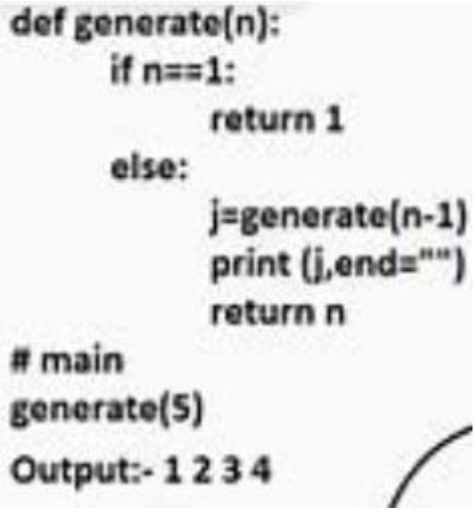
```
>>> def fact(n):
  """ Function to find factorial """
  if n == 1:
    return 1
  else:
    return (n * fact(n-1))
```



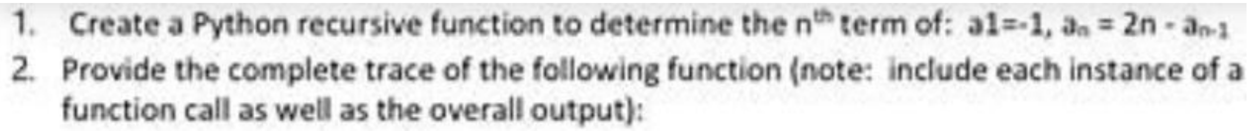
```
def binary_search(x,y):
    a = 0
    b = len(x) #If it's 'len(x)-1' here, then it's a dead loop to find the las
    c = (a+b)//2
    if y not in x. return 'Not found'
        return 'Not found'
    while x[c] ! = y:
        if x[c] < y:
            a = c
        elif x[c] > y:
            b = c
        c = (a+b)//2
    return c
```



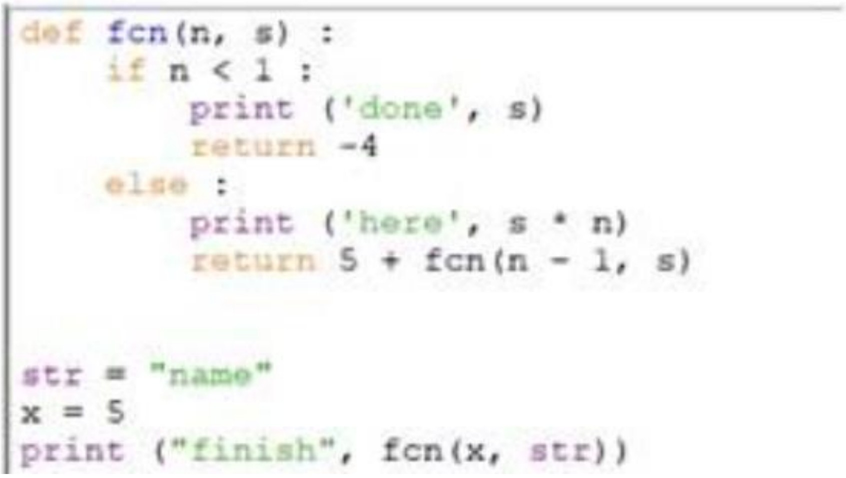
```
return 1
return 2+compute(1)
return 3+compute(2)
return 4+compute(3)
print(compute(4))
```



```
def generate(n):
        if n==1:
                return 1
        else:
                j=generate(n-1)
                print (j,end="")
                return n
# main
generate(5)
Output:- 1 2 3 4
```

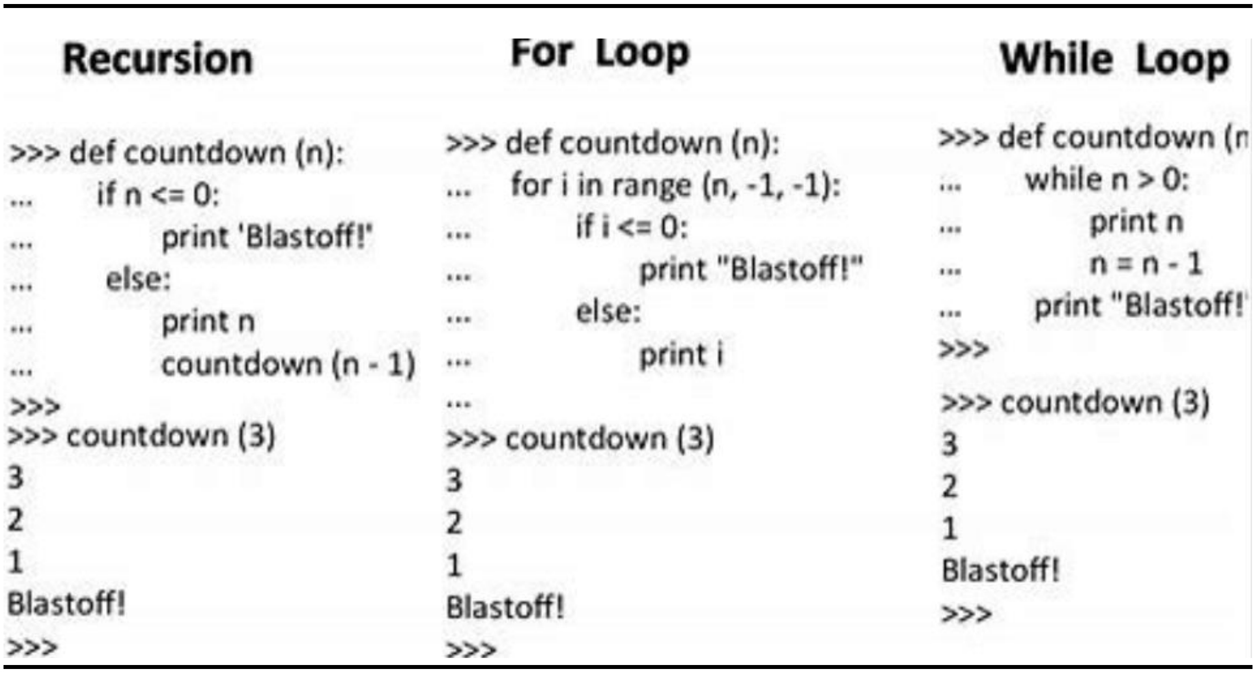


1. Create a Python recursive function to determine the n^th term of: $a1=-1$, $a_n = 2n - a_{n-1}$
2. Provide the complete trace of the following function (note: include each instance of a function call as well as the overall output):



```
def fcn(n, s) :
    if n < 1 :
        print ('done', s)
        return -4
    else :
        print ('here', s * n)
        return 5 + fcn(n - 1, s)


str = "name"
x = 5
print ("finish", fcn(x, str))
```



Recursion

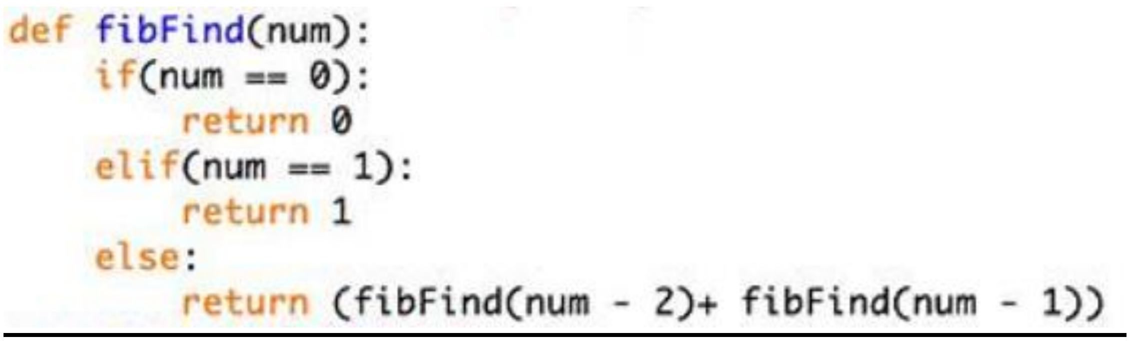
```
>>> def countdown (n):
...     if n <= 0:
...         print 'Blastoff!'
...     else:
...         print n
...         countdown (n - 1)
>>>
>>> countdown (3)
3
2
1
Blastoff!
>>>
```

For Loop

```
>>> def countdown (n):
...     for i in range (n, -1, -1):
...         if i <= 0:
...             print "Blastoff!"
...         else:
...             print i
...
>>> countdown (3)
3
2
1
Blastoff!
>>>
```

While Loop

```
>>> def countdown (n
...     while n > 0:
...         print n
...         n = n - 1
...     print "Blastoff!
>>>
>>> countdown (3)
3
2
1
Blastoff!
>>>
```



```
def fibFind(num):
    if(num == 0):
        return 0
    elif(num == 1):
        return 1
    else:
        return (fibFind(num - 2)+ fibFind(num - 1))
```