RECURSION

PROGRAMS

```
#Recursive function to find the sum of n numbers starting
from 0.(suppose we pass 6 then it process like 6+5+4+3+2+1+
0 and returns
1
3
6
10
15
21
As result

def recursion(k):
  if(k>0):
    result = k+recursion(k-1)
    print(result)
  else:
    result = 0
  return result

print("\n\nRecursion Example Results")
recursion(6)
```

```
#Recursive function to add all numbers in a list.
def sum(list):
    if len(list) == 1:
        return list[0]
    else:
        return list[0] + sum(list[1:])

print(sum([5,7,3,8,10]))
```

#Algorithm for binary search.

1. If your list is of size 0, return “not-found”.

2. Check the item located in the middle of your list.

3. If this item is equal to the item you are looking for:

you’re done! Return “found”.

4. If this item is bigger than the item you are looking for:

do a binary-search on the first half of the list.

5. If this item is smaller than the item you are looking for:

do a binary-search on the second half of the list.

#Algorithm for finding factorial recursively.

1. Take a number from the user and store it in a variable.

2. Pass the number as an argument to a recursive factorial function.

3. Define the base condition as the number to be lesser than or equal to 1 and return 1 if it is.

4. Otherwise call the function recursively with the number minus 1 multiplied by the number itself.

5. Then return the result and print the factorial of the number.

6. Exit.

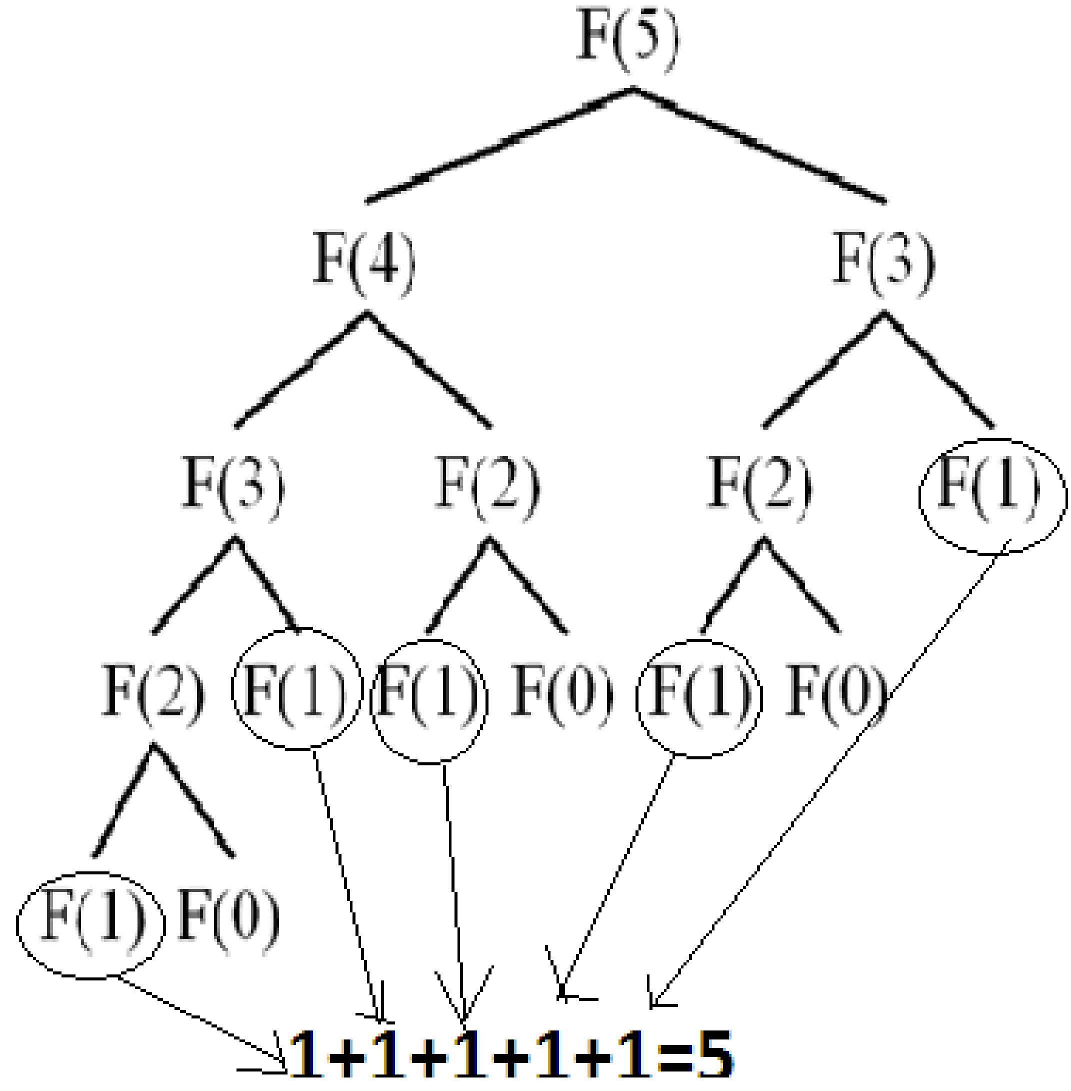
#Algorithm for finding Fibonacci number.

1. Pass a number to recursive Fibonacci function
2. If number is <=1 then return that number
3. Otherwise call the same function twice with number-1 and number-2 with addition operation

e.g. to find the 5th element of Fibonacci series,find F5 then F4 and so on till F0

$$\begin{aligned}F_2 &= F_1 + F_0 = 1 + 0 = 1\\ F_3 &= F_2 + F_1 = 1 + 1 = 2\\ F_4 &= F_3 + F_2 = 2 + 1 = 3\\ F_5 &= F_4 + F_3 = 3 + 2 = 5\end{aligned}$$

```
#Recursive python function for fibonacci series
def recur_fibo(n):
   if n <= 1:
       return n
   else:
       return(recur_fibo(n-1) + recur_fibo(n-2))
```



Term 5 is 5 fibonacci

In pseudo code, where n = 5, the following takes place:
fibonacci(4) + fibonnacci(3)

This breaks down into:
(fibonacci(3) + fibonnacci(2)) + (fibonacci(2) + fibonnacci(1))

This breaks down into:
(((fibonacci(2) + fibonnacci(1)) + ((fibonacci(1) + fibonnacci(0))) + (((fibonacci(1) + fibonnacci(0)) + 1))

This breaks down into:
((((fibonacci(1) + fibonnacci(0)) + 1) + ((1 + 0)) + ((1 + 0) + 1))
This breaks down into:
((((1 + 0) + 1) + ((1 + 0)) + ((1 + 0) + 1))