

```

1  #include<bits/stdc++.h>
2  using namespace std;
3
4  // Linear Recursion
5  /*
6  The Recursive Function call one time only
7
8      fun( n )
9      {
10         if( n > 0 )
11         {
12             -----
13             -----
14             -----
15             fun( n - 1 );
16             -----
17             -----
18             -----
19         }
20     }
21
22 */
23
24 // Tree Recursion
25 /*
26 The Recursive Function call Two time then it is call Tree Recursion
27
28     fun( n )
29     {
30         if( n > 0 )
31         {
32             -----
33             -----
34             -----
35             fun( n - 1 );
36             -----
37             -----
38             fun( n - 1 );
39             -----
40             -----
41             ----
42         }
43     }
44
45 */
46 //Tree Recursion
47 /*
48     fun( n )
49     {
50         if( n > 0 )
51         {
52             printf( n )
53

```

```

54     fun( n -1 );
55     fun( n -1 );
56 }
57 }

```

---

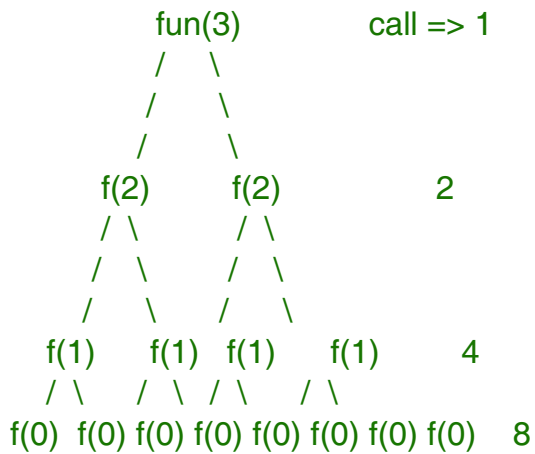
```

60     fun( 3 );

```

---

Tree Create



$$1 + 2 + 4 + 8 = 15 \text{ call}$$

This is a GP Serise:

$$2^0 + 2^1 + 2^2 + 2^3 = 2^{(3+1)} - 1 = 2^4 - 1 = 15$$

$$\text{if } n = 2^{n+1} - 1 = [ ] \text{ number of call}$$

```

84 */
85
86 void fun(int n)
87 {
88     if(n > 0)
89     {
90         cout << n << " "; //3 2 1 1 2 1 1
91         fun(n - 1);
92         fun(n - 1);
93     }
94 }
95
96
97 int main()
98 {
99     int x = 3;
100     fun(x);
101
102 }

```