

Forum: [Hack Forums](#)

Thread: [c++ recursive function](#)

Norbi34 - February 15, 2020, 10:09 am

Can someone explain me how this function returns 164618 for f(17).

```
void f(int x)
{
    if(x>3)
    {
        cout<<x-1;
        f(x/3);
        cout<<x+1;
    }
}
```

gurlhackerclub - February 15, 2020, 12:54 pm

interesting, I do notice there is no new line character or space separator.

So the function outputs, x -1, which is 17 -1
16

Then the function calls itself with 17 divided by 3. This will be 5.
So you get 5 - 1 = 4 and 5+1 = 6

Then the function adds 1 to x and outputs
18

The output would be much easier to read if you added a new line or space.
example
cout<<x-1<<" ";
16 4 6 18

SIM-3991524 - February 23, 2020, 3:33 pm

Its just the way you were using cout in f() that made it confusing

Try this example, should make a little more sense

Spoiler [\(Click to View\)](#)

Code #include <iostream>

using namespace std;

```
void f(int x, int nth_iteration)
{
    cout << nth_iteration << "\t" << x << endl;
    if (x > 3)
    {
        f(x / 3, nth_iteration + 1);
    }
}
```

```
}  
  
int main()  
{  
    f(12500, 1);  
    int done; cin >> done;  
}
```

Example output

Spoiler ([Click to View](#))

Code 1 12500

2 4166

3 1388

4 462

5 154

6 51

7 17

8 5

9 1

All f() does is divide a number by 3 until it is less than 3

saltycracker - February 26, 2020, 10:33 am

To understand recursion, you have to understand a function call. If you understand the basic mechanics of a function call then recursion is simply a function which calls itself.

Note: The topic of recursion(function recursion and recursive data structures) can get involved and very complicated but at its simplest, its really just a function call.