

# C++ Recursion

## How does recursion work?

```
void recurse()
{
    ... ..
    recurse();
    ... ..
}

int main()
{
    ... ..
    recurse();
    ... ..
}
```

The diagram illustrates the flow of recursive calls. A line from the `recurse();` statement in the `main()` function extends to the right and then turns upwards to point at the `recurse()` function definition. Another line from the `recurse();` statement inside the `recurse()` function extends to the right and then turns upwards to point at the `recurse()` function definition. The text "recursive call" is placed between these two lines, indicating the nature of the self-referencing calls.

```

7  //
8
9  #include <iostream>
10 using namespace std;
11 int n_to_the_kth_power(int n, int k)
12 {
13     if (k == 0)
14         return 1;
15     else
16         return n * n_to_the_kth_power(n, k-1);
17 }
18 int main ()
19 {
20     int n = 2, k = 10;
21     while (k >= 0) {
22         cout << n << "**" << k << " = " << n_to_the_kth_power(n,k) << endl;
23         k--;
24     }
25
26     return 0;
27 }
28
29

```

$$2^{**10} = 1024$$

$$2^{**9} = 512$$

$$2^{**8} = 256$$

$$2^{**7} = 128$$

$$2^{**6} = 64$$

$$2^{**5} = 32$$

$$2^{**4} = 16$$

$$2^{**3} = 8$$

$$2^{**2} = 4$$

$$2^{**1} = 2$$

$$2^{**0} = 1$$



```
// A C++ program to demonstrate working of  
// recursion
```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
void printFun(int test)
```

```
{
```

```
    if (test < 1)
```

```
        return;
```

```
    else
```

```
    {
```

```
        cout << test << " ";
```

```
        printFun(test-1);    // statement 2
```

```
        cout << test << " ";
```

```
        return;
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int test = 3;
```

```
    printFun(test);
```

```
}
```

output:

~~3 2 1 0 3~~