Factorial in C++

```
#include <iostream>

using namespace std;

int fact(int n) {
    if (n == 0) {
        return 1;
    } else {
        int prev = fact(n - 1);
        return n * prev;
    }
}

int main() {
    cout << fact(6) << endl;
    return 0;
}
```

Step 1: Initial Call

- Function: fact(6)
- Input: n = 6
- Condition: n != 0 → **Not base case**
- Action: Call fact(5) and calculate 6 * fact(5)

Step 2: Call fact(5)

- Function: fact(5)
- Input: n = 5
- Condition: n != 0 → **Not base case**
- Action: Call fact(4) and calculate 5 * fact(4)

Step 3: Call fact(4)

- Function: fact(4)
- Input: n = 4
- Condition: n != 0 → **Not base case**
- Action: Call fact(3) and calculate 4 * fact(3)

Step 4: Call fact(3)

- Function: fact(3)
- Input: n = 3
- Condition: n != 0 → **Not base case**
- Action: Call fact(2) and calculate 3 * fact(2)

Step 5: Call fact(2)

- Function: fact(2)
- Input: n = 2
- Condition: n != 0 → **Not base case**
- Action: Call fact(1) and calculate 2 * fact(1)

Step 6: Call fact(1)

- Function: fact(1)
- Input: n = 1
- Condition: n != 0 → **Not base case**
- Action: Call fact(0) and calculate 1 * fact(0)

Step 7: Call fact(0)

- Function: fact(0)

	• Input: n = 0
	• Condition: n == 0 → **Base case**
	• Action: Return 1
	
	Step 8: Return Values
	
	• **Return to fact(1)**:
	  ○ Calculation: 1 * fact(0) → 1 * 1 = 1
	  ○ Return: 1
	• **Return to fact(2)**:
	  ○ Calculation: 2 * fact(1) → 2 * 1 = 2
	  ○ Return: 2
	• **Return to fact(3)**:
	  ○ Calculation: 3 * fact(2) → 3 * 2 = 6
	  ○ Return: 6
	• **Return to fact(4)**:
	  ○ Calculation: 4 * fact(3) → 4 * 6 = 24
	  ○ Return: 24
	• **Return to fact(5)**:
	  ○ Calculation: 5 * fact(4) → 5 * 24 = 120
	  ○ Return: 120
	• **Return to fact(6)**:
	  ○ Calculation: 6 * fact(5) → 6 * 120 = 720
	  ○ Return: 720

Output:- 720	