CSCI 127: Joy and Beauty of Data

Lecture 12: Recursion

Reese Pearsall

Snowmester 2020

https://reesep.github.io/classes/127/main.html

Announcements

Program 3 due **tonight** @ 11:59 PM

No class tomorrow and Friday ☺

Today

Recursion, Debugging

Recursion

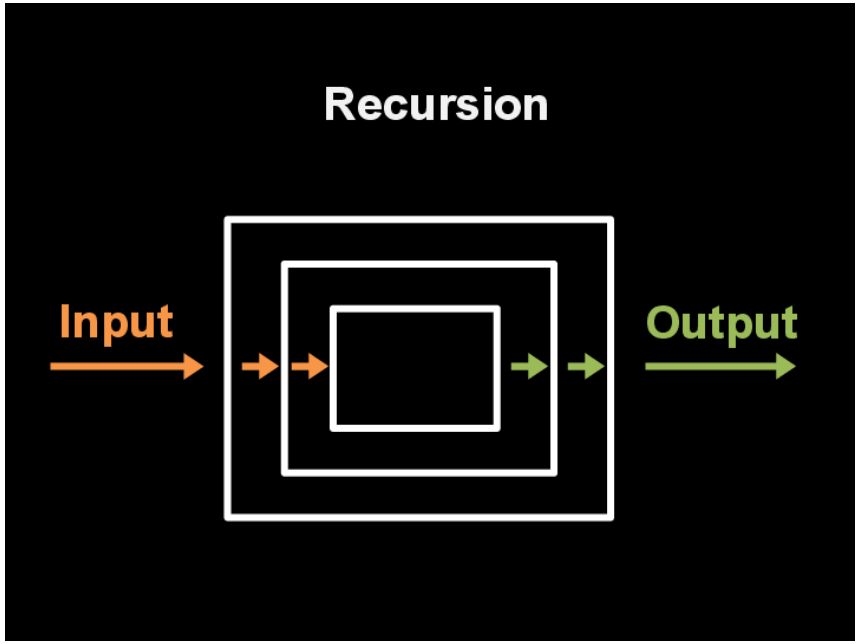
Recursion is a method of solving a problem that involves solving smaller instances of the same problem using the same function

Can be very challenging…..

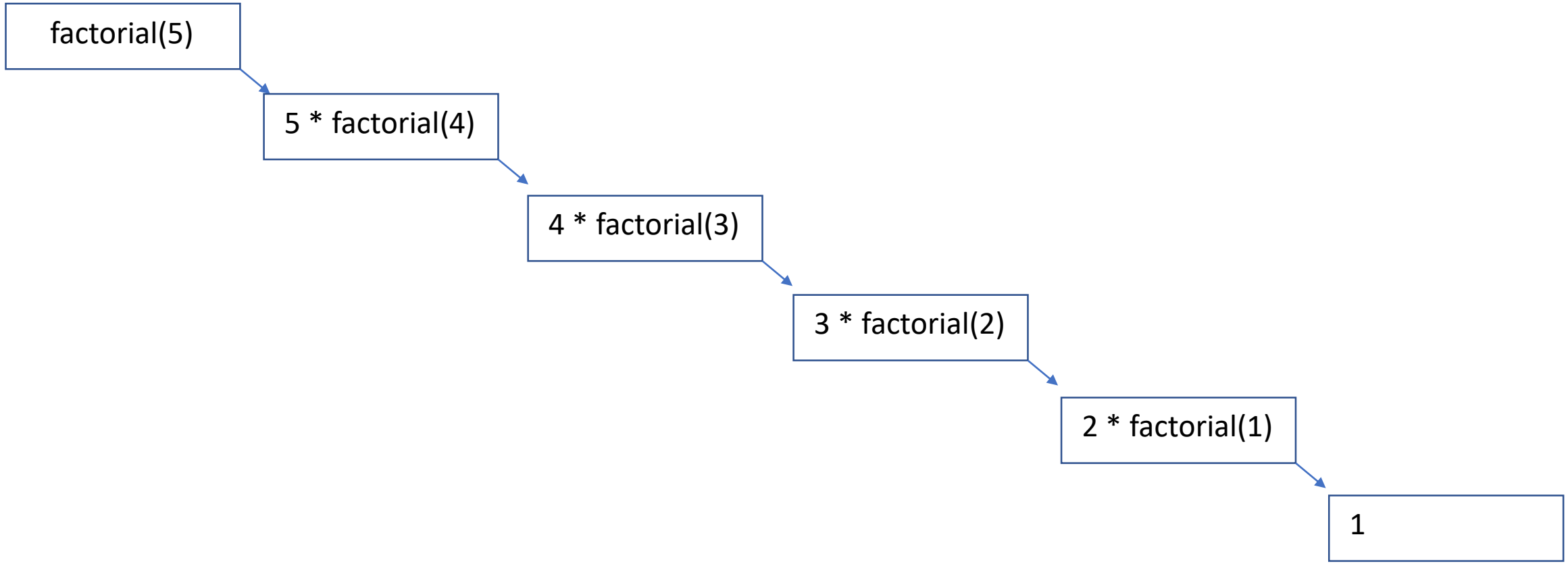
Recursion Requirements

1. Base Case
   - The “stopping point” for your recursive calls

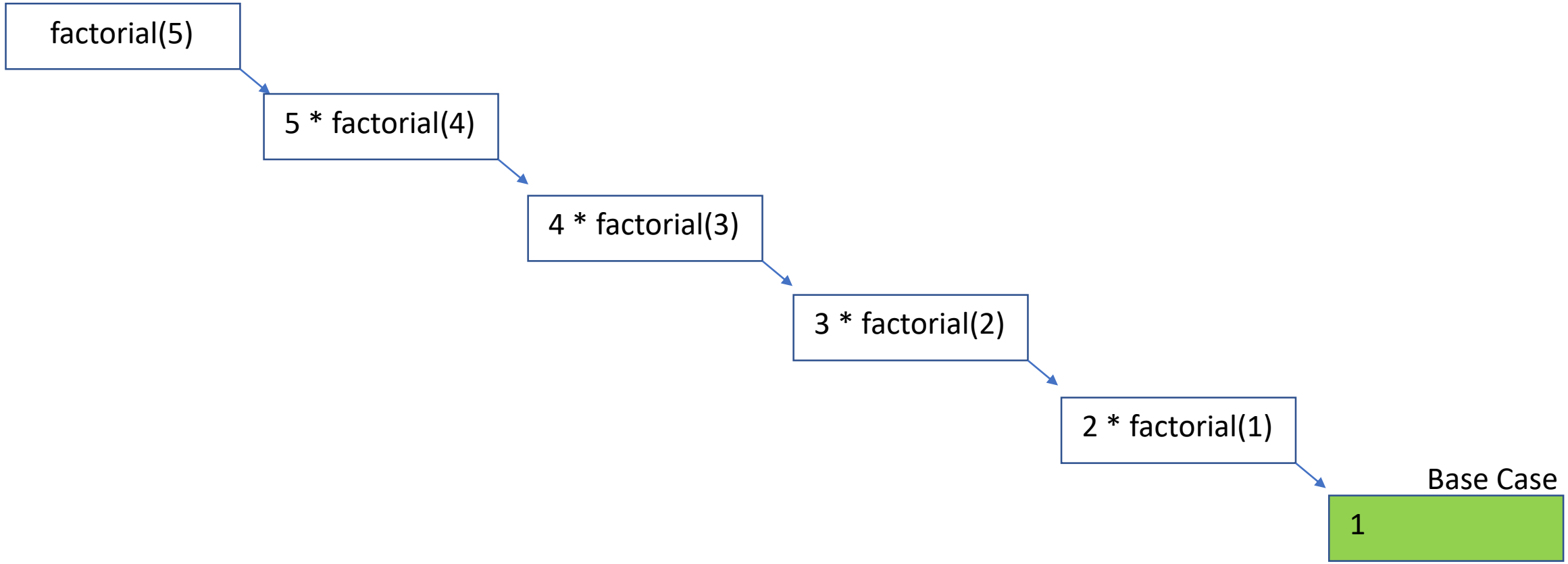
2. Recursive Case
   - Call the function again and solve a smaller problem



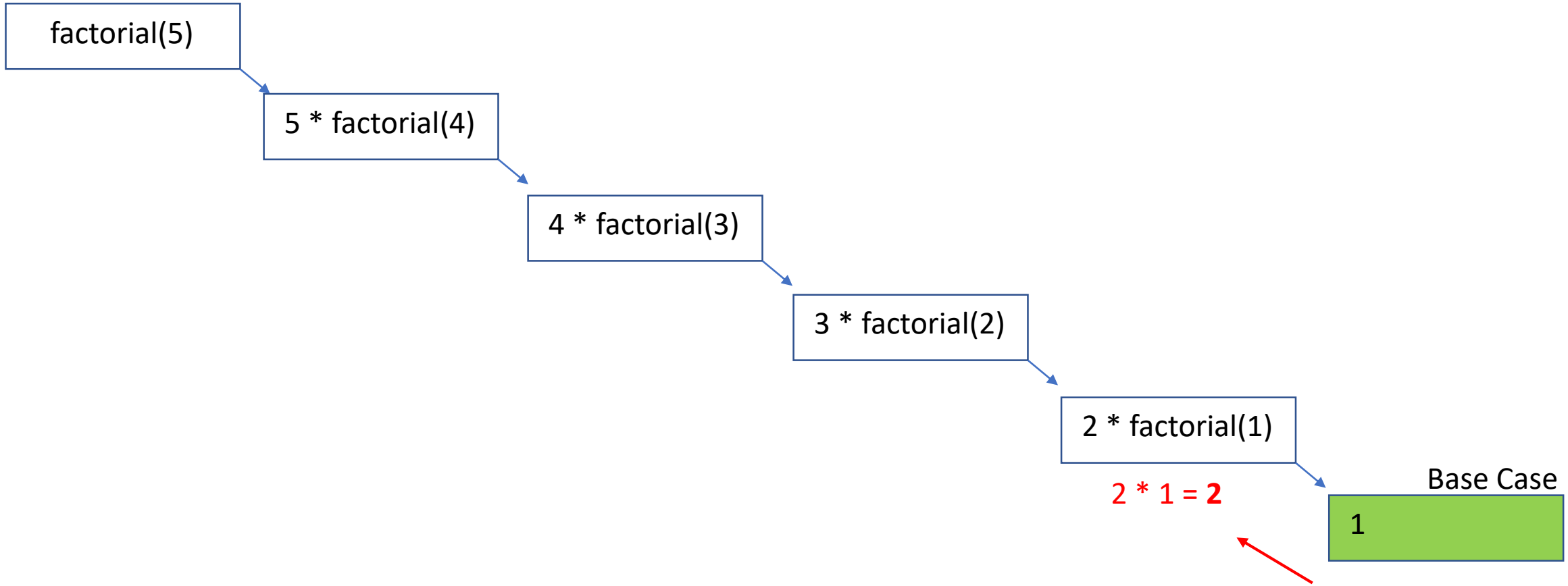
Factorial


factorial(5)
5 * factorial(4)
4 * factorial(3)
3 * factorial(2)
2 * factorial(1)
1

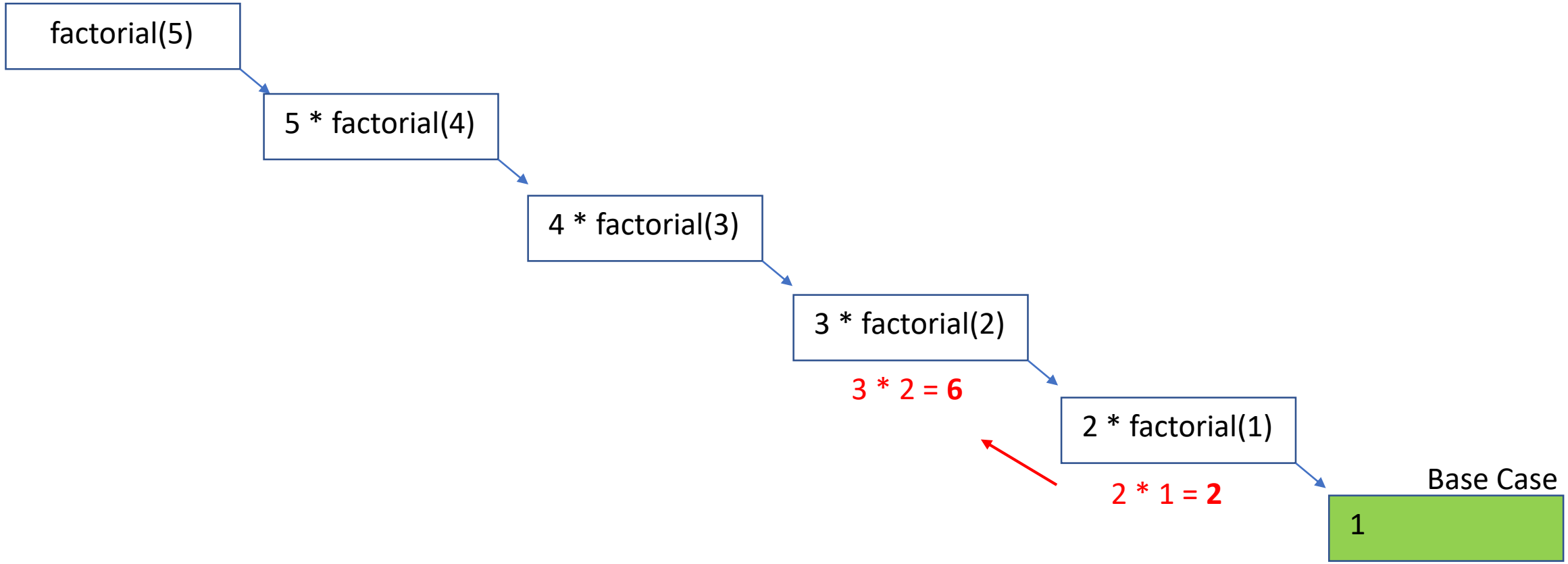
Factorial


factorial(5)
5 * factorial(4)
4 * factorial(3)
3 * factorial(2)
2 * factorial(1)
Base Case
1

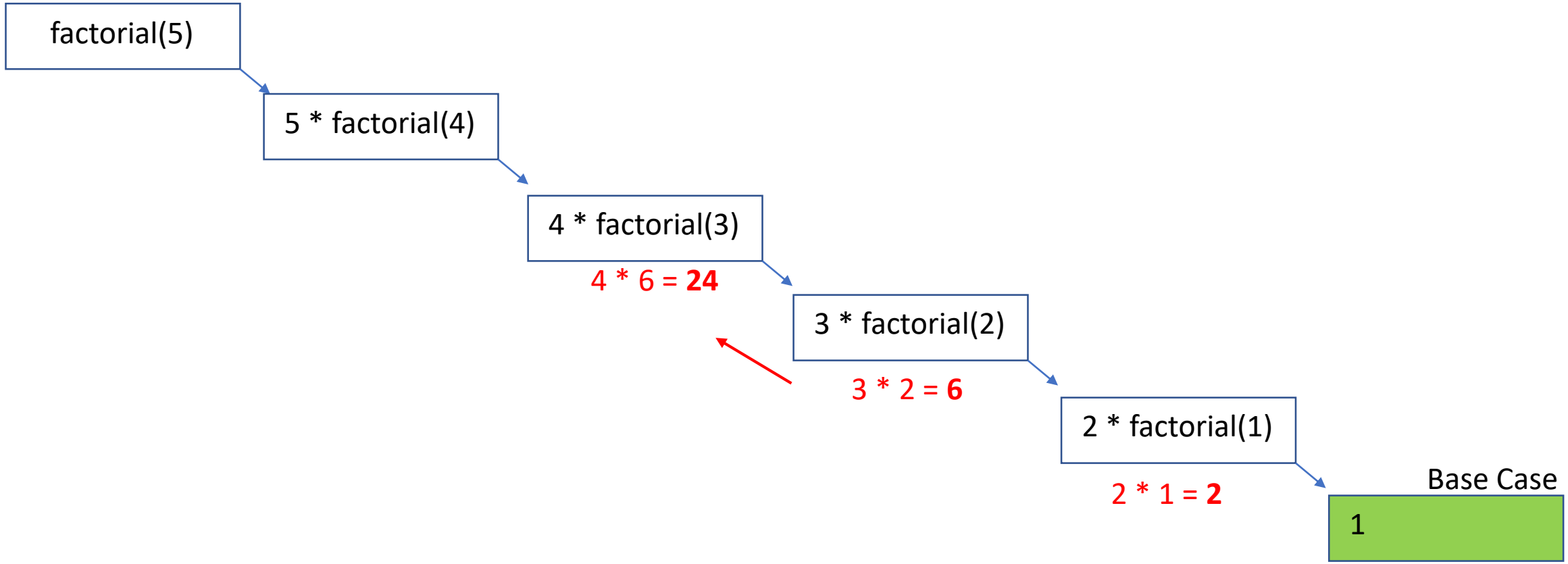
Factorial


factorial(5)
5 * factorial(4)
4 * factorial(3)
3 * factorial(2)
2 * factorial(1)
2 * 1 = 2
Base Case
1

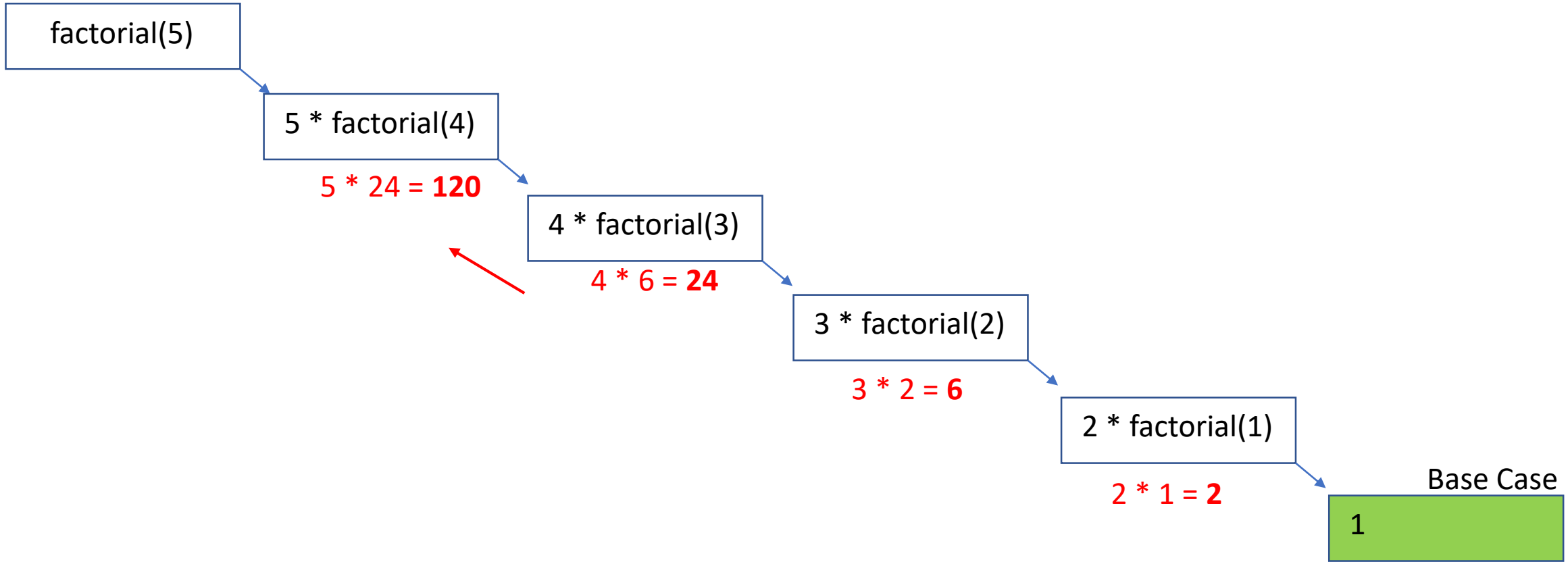
Factorial


factorial(5)
5 * factorial(4)
4 * factorial(3)
3 * factorial(2)
3 * 2 = 6
2 * factorial(1)
2 * 1 = 2
Base Case
1

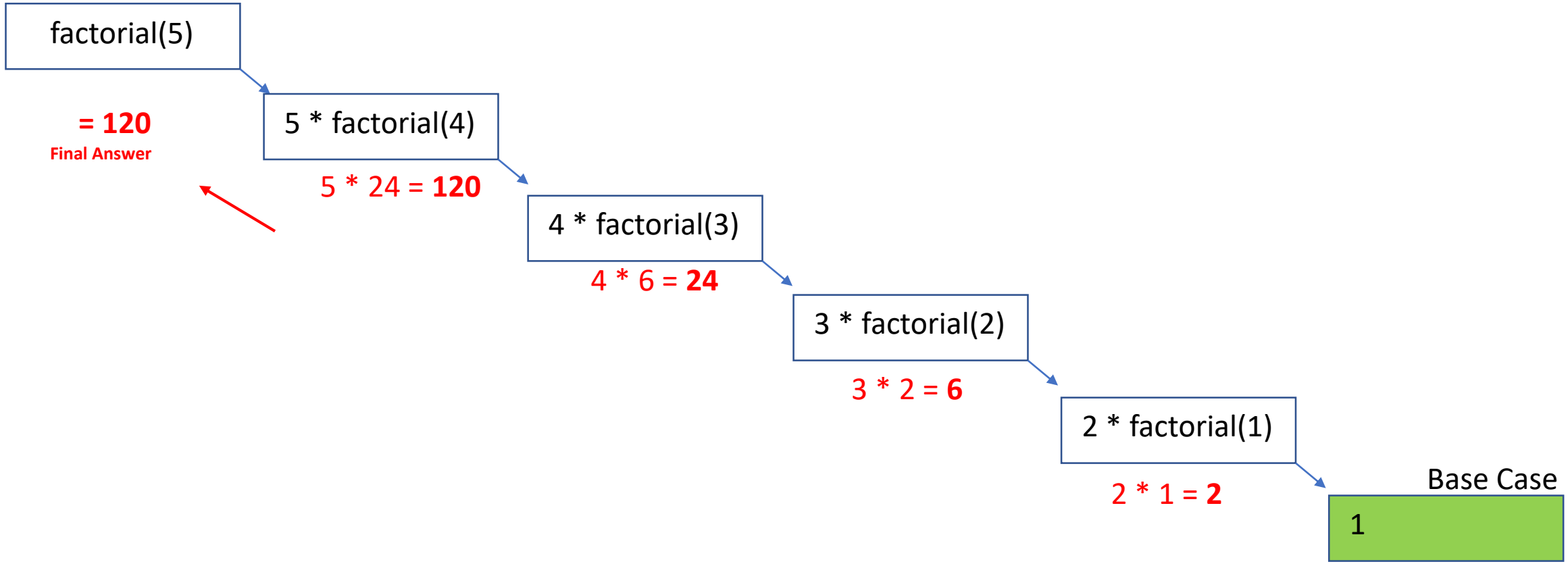
Factorial


factorial(5)
5 * factorial(4)
4 * factorial(3)
4 * 6 = 24
3 * factorial(2)
3 * 2 = 6
2 * factorial(1)
2 * 1 = 2
Base Case
1

Factorial


factorial(5)
5 * factorial(4)
5 * 24 = **120**
4 * factorial(3)
4 * 6 = **24**
3 * factorial(2)
3 * 2 = **6**
2 * factorial(1)
2 * 1 = **2**
Base Case
1

Factorial


factorial(5)
= 120
Final Answer
5 * factorial(4)
5 * 24 = 120
4 * factorial(3)
4 * 6 = 24
3 * factorial(2)
3 * 2 = 6
2 * factorial(1)
2 * 1 = 2
Base Case
1

Recursion Example

Write a recursive program that will count the number of **P**’s in a string