Solution Review: Calculate the Power of a Number Recursively

Let's go over the solution review of the challenge given in the previous lesson.

Solution#

Press the **RUN** button and see the output!

```
#include <iostream>

using namespace std;

// Recursive power function
int power(int base, int exponent) {
  // Base case
  if (exponent == 0) {
    return 1;
  }
  // Recursive case
  else
    return base * power(base, exponent - 1);
}

// main function
int main() {
  // Initialize base and exponent
  int base = 2, exponent = 4;
  // Declare variable result
  int result;
  // Call power in main and store the returned value in result
  result = power(base, exponent);
  // Print value of result
  cout << base << " raised to power " << exponent << " = " << result;
  return 0;
}
```

Explanation#

`power` function#

The recursive `power` function takes two values of type `int` in its input parameters. The first value is the `base` and the second value is the `exponent`. It returns an `int` value in the output.

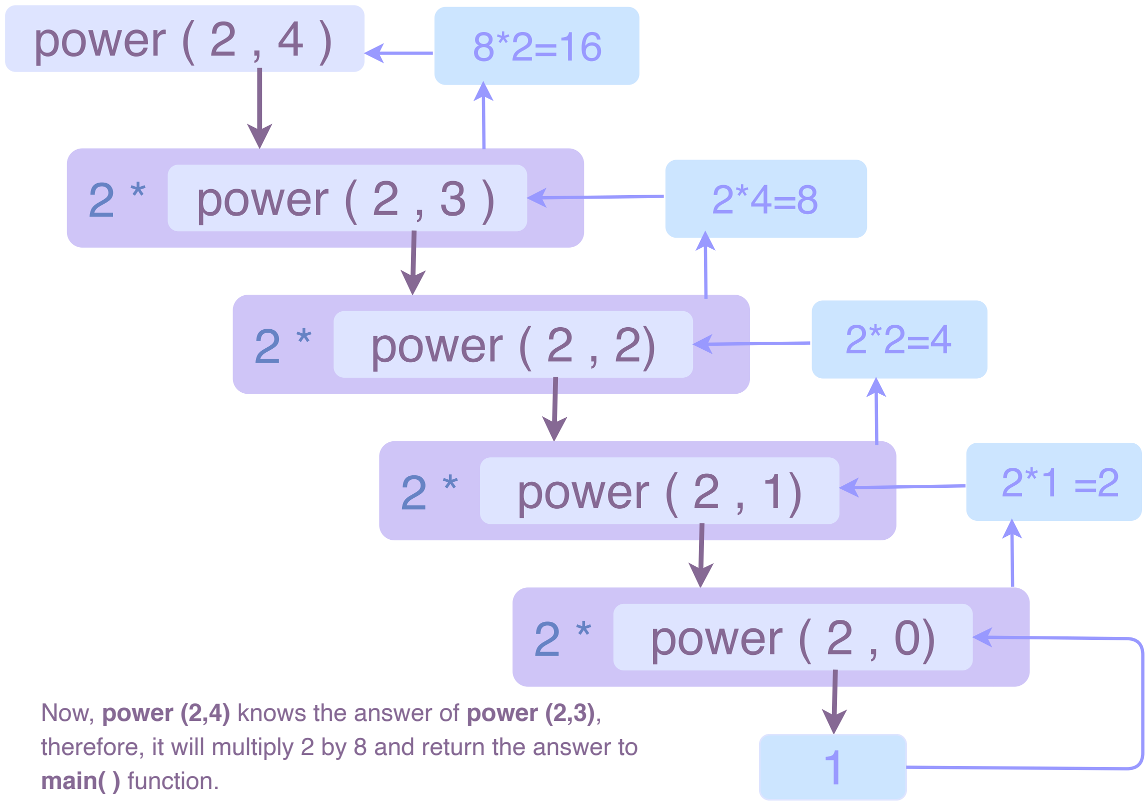
Recursive case

(/learn)

Power is calculated by multiplying the base by itself exponent times. We recursively multiply the base with the power function, each time reducing the exponent by 1. Each recursive case returns the product of `base` and `power(base, exponent-1)`.

Base case

As the exponent `0` of any number returns 1, if `exponent = 0` in our function, the function terminates after returning **1** to the calling function. There are no recursive calls in the `power` body since we have reached the simplest case. This is the base case of the `power` function.



Now, **power (2,4)** knows the answer of **power (2,3)**, therefore, it will multiply 2 by 8 and return the answer to **main( )** function.

10 of 10

← Back

Challenge 1: Calculate the Power of a ...

Next →

Challenge 2: Count the Digits in a Nu...

☑ Mark as Completed

Report an Issue