Solution Review: Count the Digits in a Number Using Recursion

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

We'll cover the following

Solution#

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

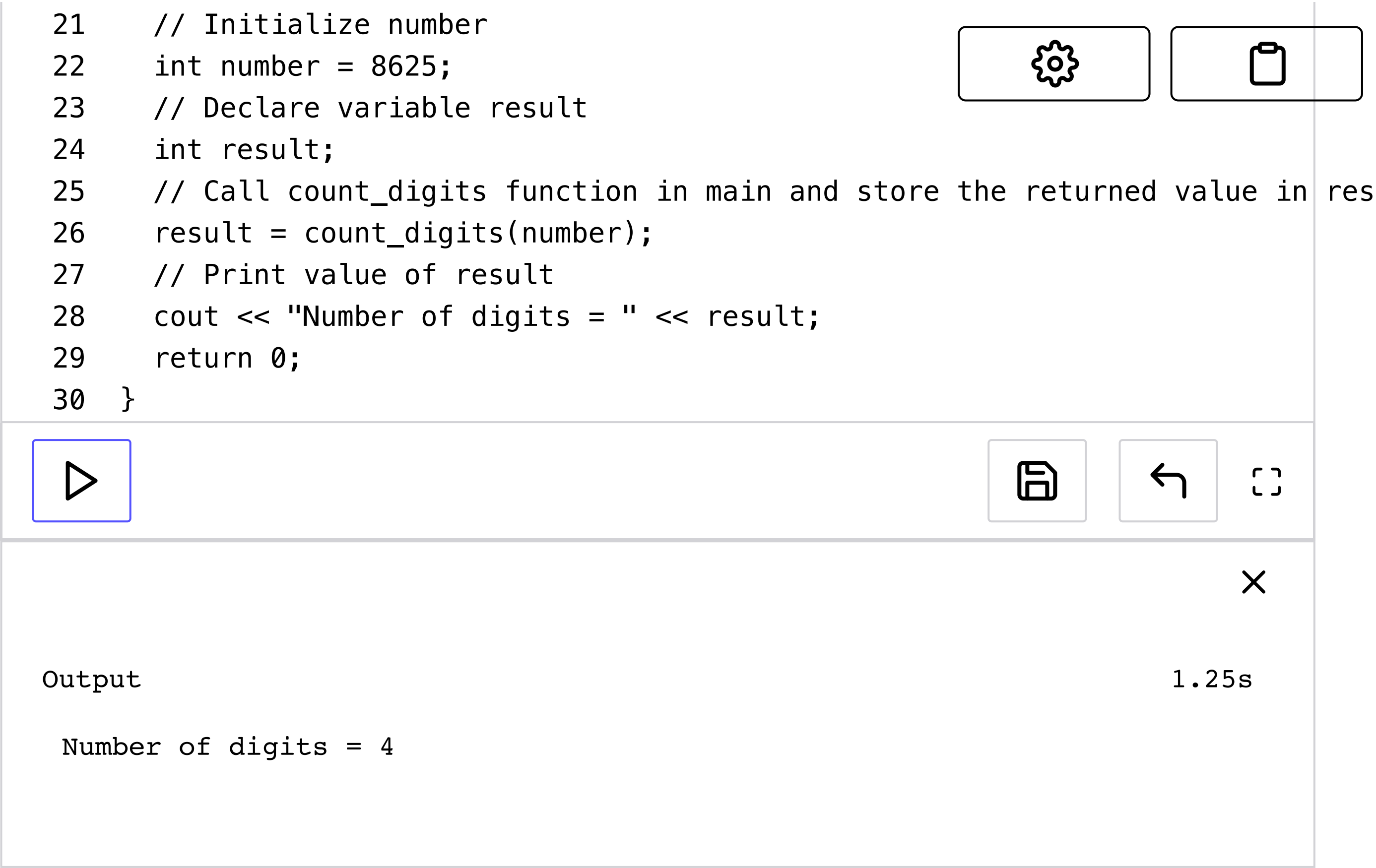
```
using namespace std;

// Recursive count_digits function
int count_digits(int number) {

  // Base Case
  if (abs(number)/10 == 0) {
    return 1;
  }
  // Recursive Case
  else {
    return 1 + count_digits(number / 10);
  }

}

// main function
int main() {
```



```
  // Initialize number
  int number = 8625;
  // Declare variable result
  int result;
  // Call count_digits function in main and store the returned value in resu
  result = count_digits(number);
  // Print value of result
  cout << "Number of digits = " << result;
  return 0;
}
```

```
Number of digits = 4
```

Explanation#

`count_digits` function#

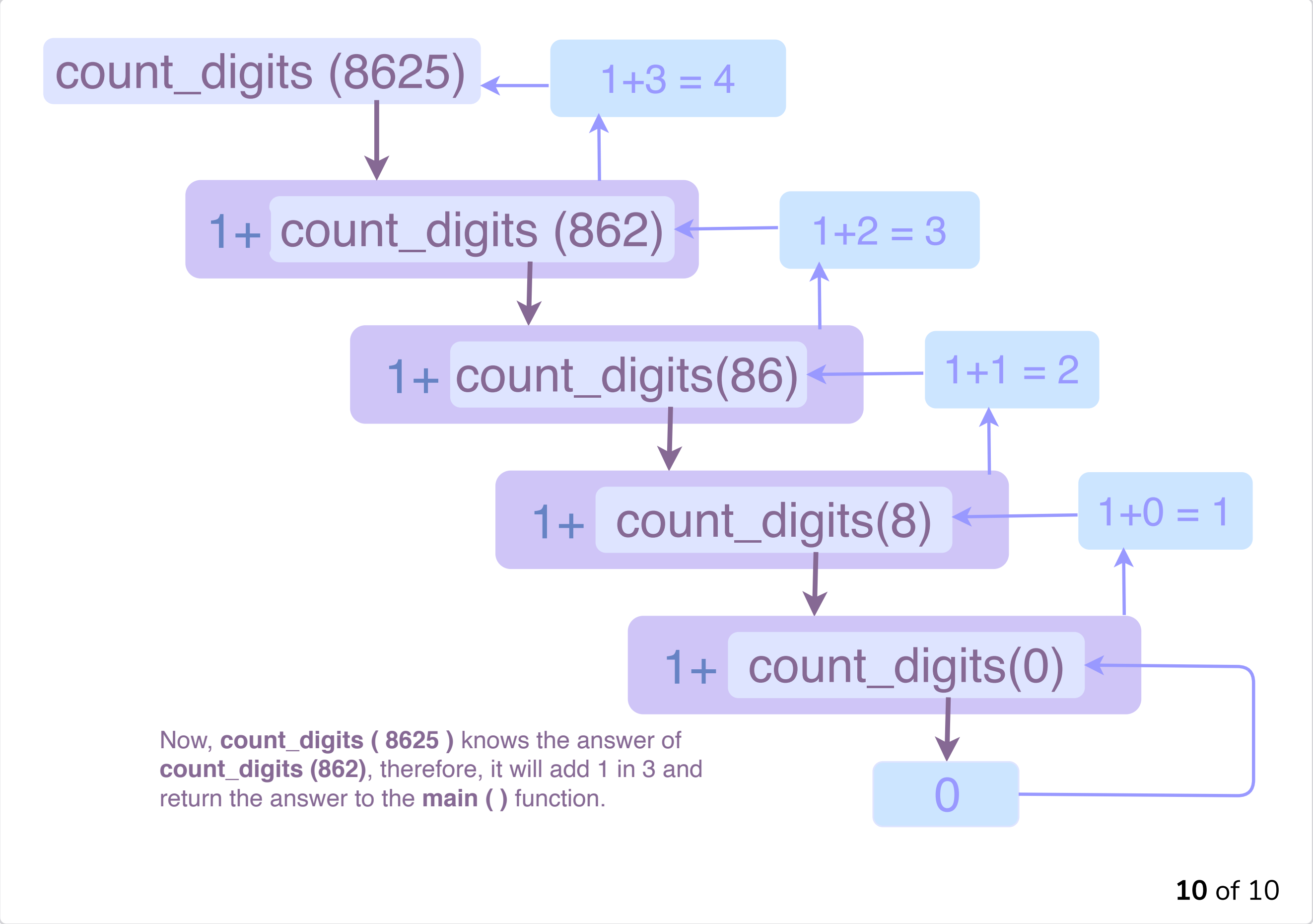
The recursive `count_digits` function takes a value of type `int` in its input parameters and returns the number of digits in the output.

Recursive case

We can count digits in a number by recursively dividing the number by 10. Each time the number is divided by 10, it loses one digit. For example, if 732 is divided by 10, it becomes 73, a two digit number from a three digit number – and we add one recursively to our return value. Return `1 + count_digits (number/10)`.

Base case

If there is only a single digit left, we return 1.



Now, **count_digits (8625)** knows the answer of **count_digits (862)**, therefore, it will add 1 in 3 and return the answer to the **main ( )** function.

10 of 10

Let’s solve another slightly more difficult challenge in the upcoming lesson.

← **Back**

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

Next →

Challenge 3: Calculate the nth Fibonac...

☑ Mark as Completed

Report an Issue