Count zeroes in C++

```
#include <iostream>
using namespace std;

int cnt = 0;

int countZerosRec(int input) {
    // Base case for initial input of 0
    if (input == 0 && cnt == 0) {
        return 1;
    }

    // Base case for recursion
    if (input == 0) {
        return cnt;
    }

    // Check if the current last digit is zero
    if (input % 10 == 0) {
        cnt++;
    }

    // Recursive call to process the next digit
    return countZerosRec(input / 10);
}

int main() {
    cout << countZerosRec(10034) << endl;
    return 0;
}
```

Dry Run of the Function

Input:

input = 10034

Step-by-Step Execution

Initial Call:

countZerosRec(10034)

- input % 10 = 4 (last digit is not 0).
- Recursive call:

  countZerosRec(1003)

Second Call:

countZerosRec(1003)

- input % 10 = 3 (last digit is not 0).
- Recursive call:

  countZerosRec(100)

Third Call:

countZerosRec(100)

- input % 10 = 0 (last digit **is** 0).
- cnt++ → cnt = 1.
- Recursive call:

  countZerosRec(10)

Fourth Call:

countZerosRec(10)

- input % 10 = 0 (last digit **is** 0).
- cnt++ → cnt = 2.
- Recursive call:

  countZerosRec(1)

Fifth Call:

countZerosRec(1)

	• input % 10 = 1 (last digit is not 0). • Recursive call: countZerosRec(0) **Base Case (Sixth Call):** countZerosRec(0) • input == 0, so return cnt. • cnt = 2. **Backtracking and Final Output:** • **Fifth Call:** Returns 2. • **Fourth Call:** Returns 2. • **Third Call:** Returns 2. • **Second Call:** Returns 2. • **Initial Call:** Returns 2.
Output:- 2	