**Sum of Digits Using Recursion**

Given a number, we need to find sum of its digits using recursion.  
Examples: 

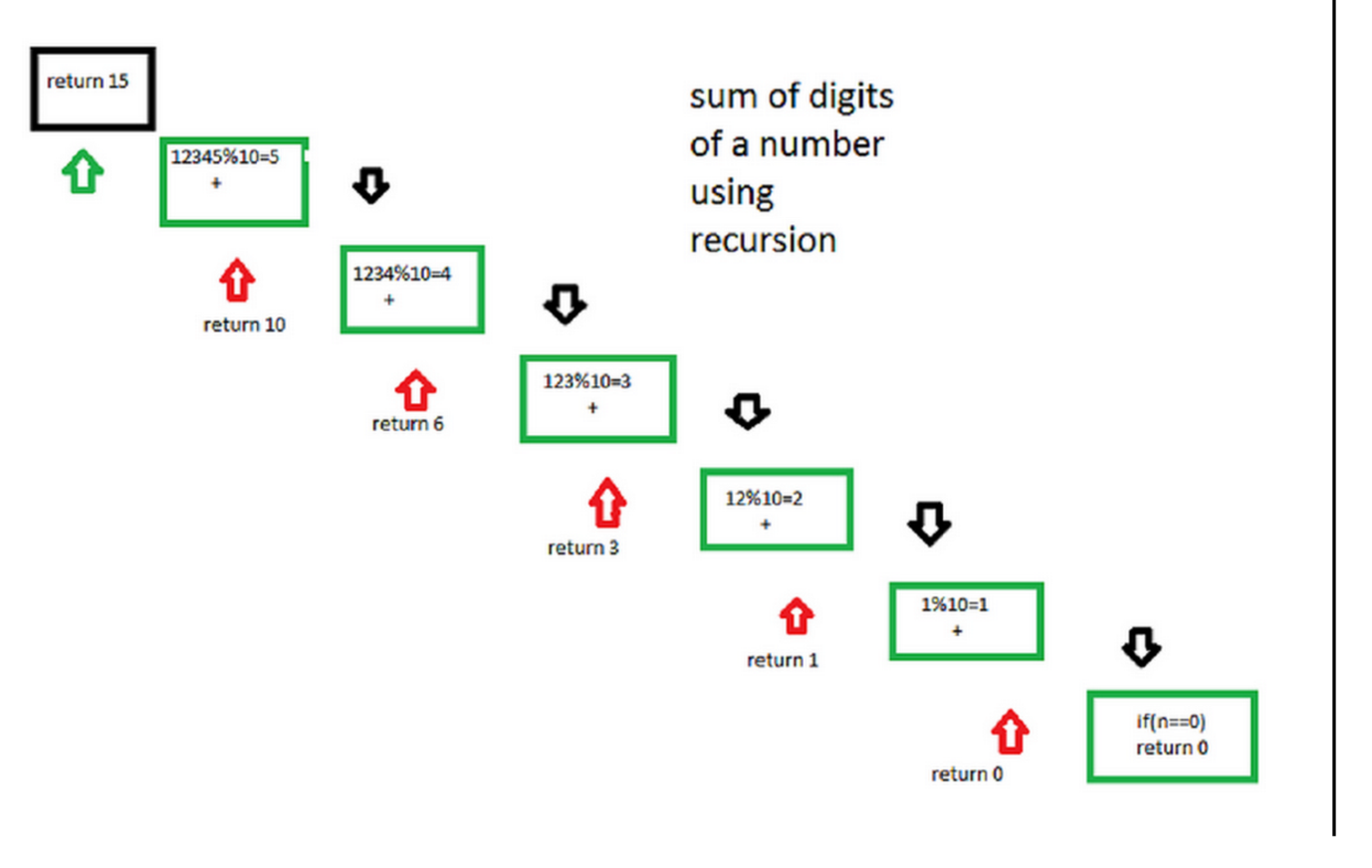
Input : 12345

Output : 15

Input : 45632

Output :20

The step-by-step process for a better understanding of how the algorithm works.   
Let the number be 12345.   
Step 1-> 12345 % 10 which is equal-too 5 + ( send 12345/10 to next step )   
Step 2-> 1234 % 10 which is equal-too 4 + ( send 1234/10 to next step )   
Step 3-> 123 % 10 which is equal-too 3 + ( send 123/10 to next step )   
Step 4-> 12 % 10 which is equal-too 2 + ( send 12/10 to next step )   
Step 5-> 1 % 10 which is equal-too 1 + ( send 1/10 to next step )   
Step 6-> 0 algorithm stops   
following diagram will illustrate the process of recursion 



C++Java

// Recursive C++ program to find sum of digits

// of a number

#include <bits/stdc++.h>

using namespace std;

// Function to check sum of digit using recursion

int sum\_of\_digit(int n)

{

if (n == 0)

return 0;

return (n % 10 + sum\_of\_digit(n / 10));

}

// Driven code

int main()

{

int num = 12345;

int result = sum\_of\_digit(num);

cout << "Sum of digits in "<< num

<<" is "<<result << endl;

return 0;

}

**Output:**

Sum of digits in 12345 is 15

Besides writing (n==0 , then return 0) in the code given above we can also write it in this manner , there will be no change in the output .

if(n<10) return n; By writing this there will be no need to call the function for the numbers which are less than 10

**Time complexity :**O(logn) where n is the given number.

**Auxiliary space :**O(logn) due to recursive stack space.