Sum of Digits using Recursion

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

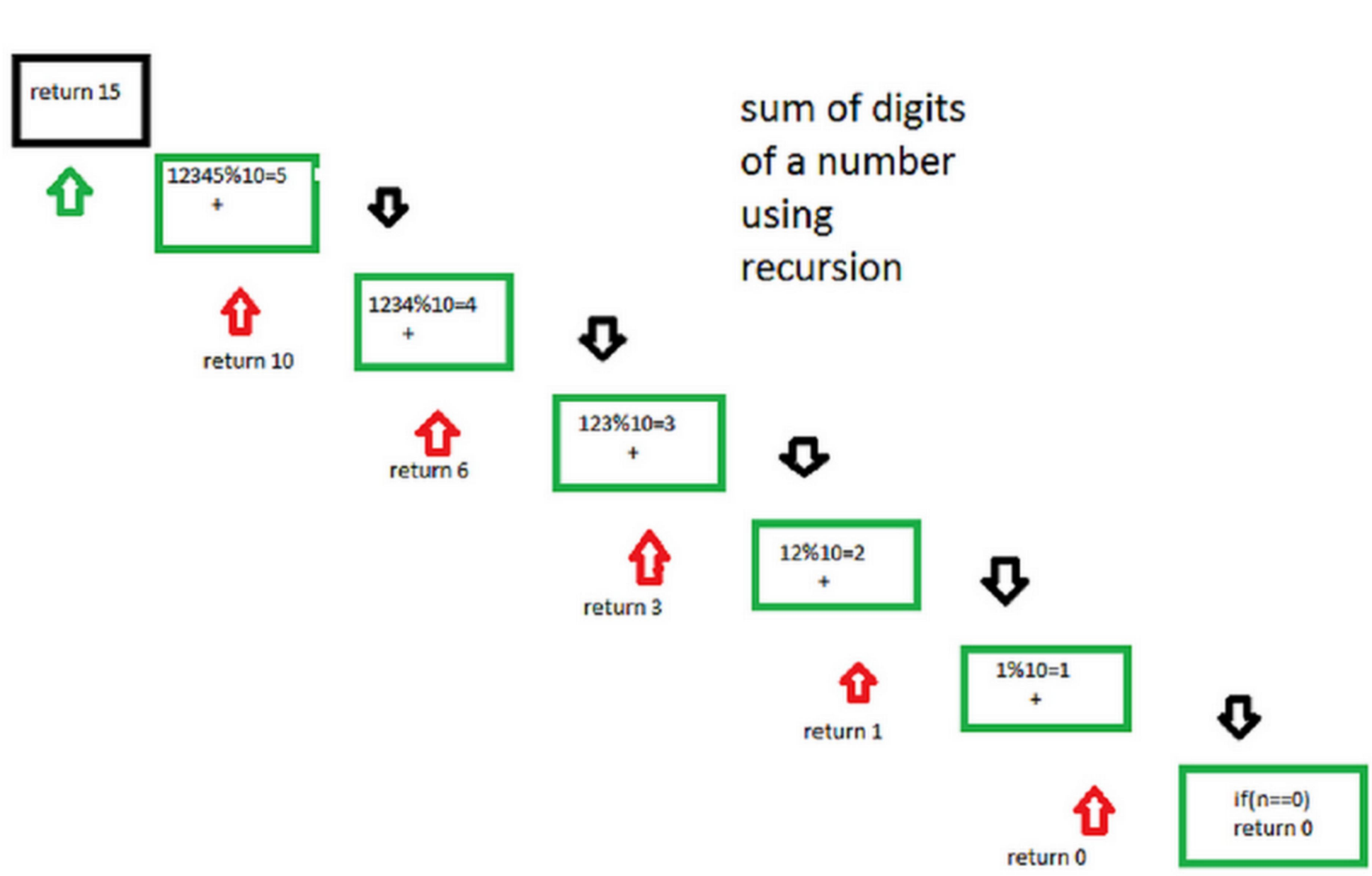
```
Input : 12345
Output : 15

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)

« Prev
Next »

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



```
# Recursive Python3 program to
# find sum of digits of a number

# Function to check sum of
# digit using recursion
def sum_of_digit( n ):
    if n == 0:
        return 0
    return (n % 10 + sum_of_digit(int(n / 10)))

# Driven code to check above
num = 12345
result = sum_of_digit(num)
print("Sum of digits in",num,"is", result)

# This code is contributed by "Sharad_Bhardwaj".
```

Output:

```
Sum of digits in 12345 is 15
```


Marked as Read

Report An Issue

If you are facing any issue on this page. Please let us know.