**Reverse a Number**

**Question:**

1. You've to display the digits of a number in reverse.
2. Take as input "n", the number for which digits have to be displayed in reverse.
3. Print the digits of the number line-wise but in reverse order.

**Input format:**

"n" where n is an integer.

**Output format:**

d1

d2

d3

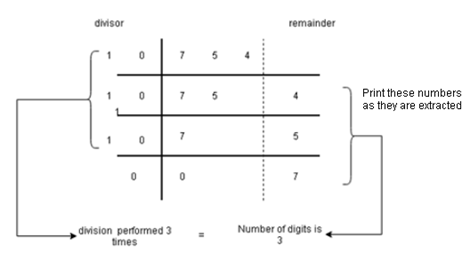
... digits of the number in reverse

**Constraints:**

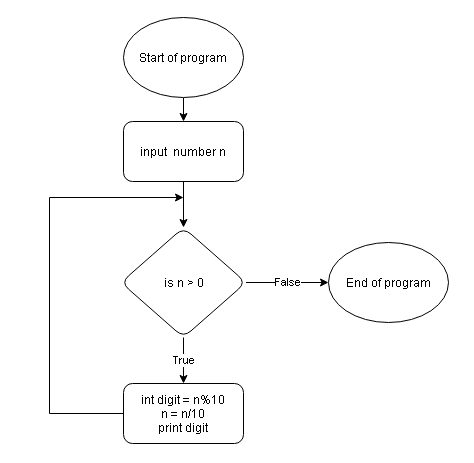
1 <= n < 10^9

**Solution Approach:**

Take 754 for instance & Keep dividing it by 10 and extracting the digit for display by storing the remainder of the division until the dividend becomes 0 or while the dividend is greater than zero.



**An algorithmic approach for the same:**



**Programming Implementation (in Java)**

import java.util.\*;

public class Main {

   public static void main(String[] args) {

       Scanner scn = new Scanner(System.in);

       int n = scn.nextInt();

       while(n != 0){

           int digit = n % 10;

           n = n / 10;

           System.out.println(digit);

       }

   }

}

**Space & Time Complexity Analysis**

* What is the **time complexity** of the above solution ?

Since, we are running a loop which reduces the number digit by digit, i.e. extracts one digit in each iteration, time complexity will be O(number of digits). Now, for a number N, number of digits will be Hence time complexity will turn out to be O(log N).

* What is the **space complexity** of above solution ?

We are not using any data structure, hence it is taking constant space. Thus space complexity will be O(1).

**Asked in Companies:** MakeMyTrip, MAQ Software