/**
 * This program looks at args[0] and, if it's a whole number, outputs
 * the sum of its digits; for example, java SumOfDigits 84735 outputs 27.
 * In this version, if args[0] is NOT a whole number, the program just
 * crashes!

 * This version also uses a value-returning method to compute the sum,
 * but the structure of that method is quite different. In particular,
 * the use of the method recurs- reappears- within the definition of the
 * method itself- also, known as recursion.
 */

public class SumOfDigits {

    public static void main ( String[] args ) {
        long number = Long.parseLong ( args[0] );
        if ( number < 0 ) { // if negative, make it positive
            number = - number;
        }
        System.out.println ( digitSum ( number ) ); // output digitSum(number)
    }

    public static long digitSum ( long n ) { // returns the sum of n's digits
        if ( n == 0 ) {
            return 0;
        } else {
            return n % 10 + digitSum(n/10); // add n's last digit plus the
                                                // digitSum of everything else
        }
    }
}