```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:45 CST 2018*/

/*Given a 32-bit signed integer, reverse digits of an integer.

        Example 1:

        Input: 123
        Output:  321
        Example 2:

        Input: -123
        Output: -321
        Example 3:

        Input: 120
        Output: 21

Note:
Assume we are dealing with an environment which could only hold integers within the 32-bit signed integer range. For the purpose of this problem, assume that your function returns 0 when the reversed integer overflows.*/


import java.lang.System;
import java.util.*;
import java.lang.Math;
import java.util.HashMap;


class ListNode {
    int val;
    ListNode next;

    ListNode(int x) {
        val = x;
    }
}


public class Main {
    public static void main(String[] args) {
        int x = -15369;

        Solution solution = new Solution();

        int receive = solution.reverse(x);


        System.out.println("haha");

    }


}


class Solution {
    public int reverse(int x) {


        int result = 0;
        int currentResult = 0;
        int tail = 0;

        while (x != 0) {
            tail = x % 10;

            currentResult = result * 10 + tail;

            if (currentResult / 10 != result) {
                return 0;
            }
```

```
72                  result = currentResult;
73                  x /= 10;
74              }
75
76              return result;
77          }
78      }
```