Reverse Integer

Reverse digits of an integer.

```
Example1: x = 123, return 321
Example2: x = -123, return -321
```

click to show spoilers.

Have you thought about this?
Here are some good questions to ask before coding. Bonus points for you if you have already thought through this!

If the integer's last digit is 0, what should the output be? ie, cases such as 10, 100.

Did you notice that the reversed integer might overflow? Assume the input is a 32-bit integer, then the reverse of 1000000003 overflows. How should you handle such cases?

For the purpose of this problem, assume that your function returns 0 when the reversed integer overflows.

Update (2014-11-10):
Test cases had been added to test the overflow behavior.

Solution 1

Only 15 lines. If overflow exists, the new result will not equal previous one. No flags needed. No hard code like 0xf7777777 needed. Sorry for my bad english.

```
public int reverse(int x)
{
    int result = 0;

    while (x != 0)
    {
        int tail = x % 10;
        int newResult = result * 10 + tail;
        if ((newResult - tail) / 10 != result)
        { return 0; }
        result = newResult;
        x = x / 10;
    }

    return result;
}
```

written by bitzhuwei original link here

Solution 2

```
public int reverse(int x) {
        long rev= 0;
        while( x != 0){
            rev= rev*10 + x % 10;
            x= x/10;
            if( rev > Integer.MAX_VALUE || rev < Integer.MIN_VALUE)
                return 0;
        }
        return (int) rev;
    }
```

written by kbakhit original link here

Solution 3

Throw an exception? Good, but what if throwing an exception is not an option? You would then have to re-design the function (ie, add an extra parameter).

written by caidiexunmeng original link here