1. 题目
2. 题干

给出一个 32 位的有符号整数，你需要将这个整数中每位上的数字进行反转。

注意:

假设我们的环境只能存储得下 32 位的有符号整数，则其数值范围为 [−231,  231 − 1]。请根据这个假设，如果反转后整数溢出那么就返回0。

1. 示例

示例 1:

输入: 123

输出: 321

 示例 2:

输入: -123

输出: -321

示例 3:

输入: 120

输出: 21

二．思路

反转整数，这种做过很多次了，但是这次要限定在 [−231,  231− 1]范围内，因此需要考虑反转后的正数或者负数是否依然在这个范围内。

当整数溢出该范围时，即正数大于整数最大值MAX\_VALUE，负数要小于整数的最小值MIN\_VALUE。当我们要将整数的个位数弹出时设为pop，反转后的整数设为ans。

则

正数溢出条件： ans \* 10 + pop > MAX\_VALUE

则，当ans > MAX\_VALUE / 10时必定溢出，或者

Ans == MAX\_VALUE / 10且 pop > 7时，必定溢出。（7 是 MAX\_VALUE的个位数）

同理，负数溢出条件：

Ans \* 10 + pop < MIN\_VALUE

当 ans < MIN\_VALUE / 10 时必定溢出 ，或者

Ans = MIN\_VALUE / 10 且 pop < -8 时，必定溢出。（8是MIN\_VALUE的个位数）

三．代码实现

#include<string>

using namespace std;

class Solution {

public:

int reverse(int x) {

//结果值

int ans = 0;

//如果为0输出0

while (x != 0) {

//弹出个位数

int pop = x % 10;

//剩余的数

x /= 10;

//正数时，判断溢出条件，INT\_MAX是int32正数最大值

if (ans > INT\_MAX / 10 || (ans == INT\_MAX / 10 && pop > 7))

return 0;

//负数时，判断溢出条件，INT\_MIN是int32负数最小值

if (ans < INT\_MIN / 10 || (ans == INT\_MIN && pop < -8))

return 0;

ans = ans \* 10 + pop;

}

return ans;

}

};