7. Reverse Integer

Easy

19602901FavoriteShare

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 store integers within the 32-bit signed integer range: [−231, 231 − 1]. For the purpose of this problem, assume that your function returns 0 when the reversed integer overflows.

class Solution {

public:

int reverse(int x) {

string s=to\_string(x);

int first=(s[0]=='-')?1:0;

int last=s.size()-1;

char temp;

while(first<last){

temp=s[first];

s[first]=s[last];

s[last]=temp;

first++;

last--;

}

int ret=0;

try{

ret=stoi(s);

}catch(const std::out\_of\_range& oor){

return 0;

}

return ret;

}

};

Success

[Details](https://leetcode.com/submissions/detail/213302494/)

Runtime: 12 ms, faster than 98.25% of C++ online submissions for Reverse Integer.

Memory Usage: 14.6 MB, less than 5.06% of C++ online submissions forReverse Integer.