# 8. String to Integer (atoi)

Implement the myAtoi(string s) function, which converts a string to a 32-bit signed integer.

The algorithm for myAtoi(string s) is as follows:

Whitespace: Ignore any leading whitespace (" ").

Signedness: Determine the sign by checking if the next character is '-' or '+', assuming positivity is neither present.

Conversion: Read the integer by skipping leading zeros until a non-digit character is encountered or the end of the string is reached. If no digits were read, then the result is 0.

Rounding: If the integer is out of the 32-bit signed integer range [-231, 231 - 1], then round the integer to remain in the range. Specifically, integers less than -231 should be rounded to -231, and integers greater than 231 - 1 should be rounded to 231 - 1.

Return the integer as the final result.

## SOLUTION IN C++

class Solution {

public:

int myAtoi(string s) {

trim(s);

if (s.empty())

return 0;

const int sign = s[0] == '-' ? -1 : 1;

if (s[0] == '+' || s[0] == '-')

s = s.substr(1);

long num = 0;

for (const char c : s) {

if (!isdigit(c))

break;

num = num \* 10 + (c - '0');

if (sign \* num < INT\_MIN)

return INT\_MIN;

if (sign \* num > INT\_MAX)

return INT\_MAX;

}

return sign \* num;

}

private:

void trim(string& s) {

s.erase(0, s.find\_first\_not\_of(' '));

s.erase(s.find\_last\_not\_of(' ') + 1);

}

};