

myAtoi(string s) problem

We need to simulate a manual string-to-integer parser that handles:

- whitespace trimming
- optional sign
- digit reading
- overflow clamping

[Edge Cases:]

Input is only whitespace: \rightarrow return 0

Input is "+0123" \rightarrow stop after 0, return 0

Input is "-0001234" \rightarrow treat it as -1234

Input starts with non-digit after optional sign (like "-abc") \rightarrow return 0

[Implementation:]

1. Trim leading whitespace: use a pointer / index to skip all spaces at the beginning. Don't use built-in trimming.

2. Check for sign: · If you see '-', set sign to negative.
· If you see '+', set sign to positive.
· If neither, default is positive.
· Advance the index only if '+' or '-' is found.

3. Read digits: Loop as long as current character is a digit ('0' to '9').
For each digit: · convert it using $\text{digit} = s[i] - '0'$
· before adding to result, check if multiplying result by 10 and adding digit will overflow 32-bit signed range.

Use: $\text{if}(\text{result} > \text{INT_MAX}/10 \parallel (\text{result} == \text{INT_MAX}/10 \ \&\& \text{digit} > 7)) \parallel \text{clamp}$

Clamp the result: · If overflowing positive, return INT_MAX
· If overflowing negative, return INT_MIN

4. Multiply and Accumulate: $\text{result} = \text{result} * 10 + \text{digit};$

5. Apply Sign and Return : After the loop, return sign & result.