150. Evaluate Reverse Polish Notation

Medium

444318FavoriteShare

Evaluate the value of an arithmetic expression in [Reverse Polish Notation](http://en.wikipedia.org/wiki/Reverse_Polish_notation).

Valid operators are +, -, \*, /. Each operand may be an integer or another expression.

Note:

* Division between two integers should truncate toward zero.
* The given RPN expression is always valid. That means the expression would always evaluate to a result and there won't be any divide by zero operation.

Example 1:

Input: ["2", "1", "+", "3", "\*"]  
Output: 9  
Explanation: ((2 + 1) \* 3) = 9

Example 2:

Input: ["4", "13", "5", "/", "+"]  
Output: 6  
Explanation: (4 + (13 / 5)) = 6

Example 3:

Input: ["10", "6", "9", "3", "+", "-11", "\*", "/", "\*", "17", "+", "5", "+"]  
Output: 22  
Explanation:   
 ((10 \* (6 / ((9 + 3) \* -11))) + 17) + 5  
= ((10 \* (6 / (12 \* -11))) + 17) + 5  
= ((10 \* (6 / -132)) + 17) + 5  
= ((10 \* 0) + 17) + 5  
= (0 + 17) + 5  
= 17 + 5  
= 22

class Solution {

public:

int evalRPN(vector<string>& tokens) {

stack<int> stk;

for(int i=0;i<tokens.size();i++){

if(isdigit(tokens[i][0])||isdigit(tokens[i][1])) stk.push(stoi(tokens[i]));

else{

int a = stk.top();

stk.pop();

int b = stk.top();

stk.pop();

if(tokens[i]=="+") stk.push(a+b);

else if(tokens[i]=="-") stk.push(b-a);

else if(tokens[i]=="\*") stk.push(a\*b);

else stk.push(b/a);

}

}

return stk.top();

}

};

Success

[Details](https://leetcode.com/submissions/detail/211398149/)

Runtime: 16 ms, faster than 100.00% of C++ online submissions for Evaluate Reverse Polish Notation.

Memory Usage: 11.4 MB, less than 87.13% of C++ online submissions for Evaluate Reverse Polish Notation.