Problem #150 : Evaluate Reverse Polish Notation

<https://leetcode.com/problems/evaluate-reverse-polish-notation/>

Solution:

<https://leetcode.com/problems/evaluate-reverse-polish-notation/discuss/839458/Simple-Python-3-Solution-using-stack-Runtime-beats-81.75>

1. Used "from collections import deque" for implementing a stack called my\_stack.
2. Make a list of operators
3. Iterate through the tokens.
4. If the token is not an operator (check whether it belongs to the operators list), then push it on the stack (i.e. append to my\_stack.
5. If the tolem os am operator, pop the last 2 numbers from the stack and apply the approriate operation. Then push the answer into the stack again.
6. After the for loop ends, if stack is not empty, pop the stack and return the number stored in the stack.

```

from collections import deque

class Solution:

def evalRPN(self, tokens: List[str]) -> int:

n = len(tokens)

my\_stack = deque()

operators = ["+", "-", "\*", "/"]

for i in range(n):

if tokens[i] not in operators:

my\_stack.append(tokens[i])

else:

num1 = int(my\_stack.pop())

num2 = int(my\_stack.pop())

if tokens[i] == "+":

my\_stack.append(num2 + num1)

elif tokens[i] == '-':

my\_stack.append(num2 - num1)

elif tokens[i] == '\*':

my\_stack.append(num2 \* num1)

elif tokens[i] == '/':

my\_stack.append(num2 / num1)

if my\_stack:

return(int(my\_stack.pop()))

else:

return

```