https://course.acciojob.com/idle?question=1cf36385-8217-44a1-95b7-6866e0346f4c

- MEDIUM
- Max Score: 40 Points

Evaluate Reverse Polish Notation

Evaluate the value of an arithmetic expression in Reverse Polish Notation.

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

Note that division between two integers should truncate toward zero.

It is guaranteed that the given RPN expression is always valid. That means the expression would always evaluate to a result, and there will not be any division by zero operation.

Input Format:

First line containing 'n', number of string the array

Next 'n' lines contain the elements of the tokens array.

Output Format:

Single integer denoting the result of the reverser polish notation.

Example 1

Input

```
5
3
1
+
4
*
```

Output

```
16
```

Explanation

```
((3 + 1) * 4) = 16
```

Example 2

Input

```
5
5
21
5
/
+
```

Output

```
9
```

Explanation

```
(5 + (21 / 5)) = 9
```

Constraints

```
1 <= tokens.length <= 10^4
```

```
tokens[i] is either an operator: "+", "-", "*", or "/", or an integer in the range [-200,
200]
```

Topic Tags

- **Math**
- **Stacks**
- **Arrays**

My code

```
// n java
import java.util.*;

public class Main
{
  static int evalRPN(String[] tokens) {
    //Write your code here
        Stack<Integer>st=new Stack<>();
        for(String s:tokens)
            {
                if("+".equals(s))
                {
                    int t=st.pop();
                    int p=st.pop();
                    st.push(t+p);
                }

            else    if("-".equals(s))
                {
                    int t=st.pop();
                    int p=st.pop();
                    st.push(-t+p);
                }
```

```
          else  if("/".equals(s))
              {
                  int t=st.pop();
                  int p=st.pop();
                  st.push((int)(1D/t*p));
              }

          else  if("*".equals(s))
              {
                  int t=st.pop();
                  int p=st.pop();
                  st.push(t*p);
              }
              else
              {
                  st.push(Integer.valueOf(s));
              }
          }
      return st.pop();
}
  public static void main(String[] args) {
   Scanner input = new Scanner(System.in);
   int n = input.nextInt();
   String[] arr = new String[n];
   for(int i=0; i<n; i++)
   {
     arr[i] = input.next();
   }
   System.out.println(evalRPN(arr));
```

```
        }
    }
```