

```
1  /* Problem 2--APL Expressions
2     In fact, APL expressions are easier to compute than normal
3     used in most languages.  You just have to process the string one
4     character at a time. */
5
6  import java.io.*;
7  import java.util.*;
8
9  public class prob2 {
10
11     private static Scanner in;
12     private static PrintWriter out;
13
14     private static String expr;
15     private static int cs;
16
17     public static void main (String[] args) throws Exception {
18
19         in = new Scanner (new File ("prob2.in"));
20         out = new PrintWriter ("prob2.out");
21         cs = 1;
22         while (!(expr = in.nextLine ().equals ("")) {
23             String exprcpy = expr; //read in the string
24             out.print ("Case "+(cs++)+": "+exprcpy+" evaluates to "+Process ()
25                 +".\r\n\r\n");
26         }
27         in.close ();
28         out.close ();
29     }
30
31     /* Process processes the input string one character at a time and
32        prints the answer. */
33     public static int Process () throws Exception {
34
35         int ans;
36         if (expr.charAt(0)=='(') {
37             expr = expr.substring(1); //Remove "(" if it's there
38             ans = Process (); //Process the expression inside the ()'s
39         } else {
40             ans = 0;
41             while (!(expr.equals("") && expr.charAt(0) >= '0' &&
42                 expr.charAt(0) <= '9') { //pull off digits to append
43                 ans = 10*ans+(expr.charAt(0)-'0');
44                 expr = expr.substring (1);
45             }
46         }
47         if (expr.equals("")) return ans; //If the string contained just
48         if (expr.charAt(0)=='') { //a number, then that's the answer.
49             expr = expr.substring (1);
50             return ans;
51         }
52         char op = expr.charAt(0); //Otherwise, pull off an operator
53         expr = expr.substring (1);
54         int ans2 = Process (); //Recursively compute the right side
55         switch (op) {
56             case '+' : return ans+ans2; //Apply the operator
57             case '-' : return ans-ans2;
58             case '*' : return ans*ans2;
59         }
60         return 0;
61     }
62
63
64 }
65
```