Daily Work Practice Date: 11/9/2023

Program: Sum root to leaf numbers

```
package com.ishwarchavan;
import java.util.*;

public class TreeNode {      //class created
static class Node {        //A Binary tree node
      int data;
      Node left, right;

      Node(int val) {     // Constructor to create a new node
      data = val;
      left = right = null;
      }
};

static void treePathsSumUtil(Node root, ArrayList<String>
currPath,ArrayList<ArrayList<String> > allPath) {

      if (root == null)    // Base Case
      return;
      currPath.add(("" + root.data));   // append the root data in string format in
currPath

      if (root.left == null && root.right == null)     // if we found a leaf node we
copy the currPath to allPath
      allPath.add(new ArrayList<>(currPath));

      treePathsSumUtil(root.left, currPath, allPath);    // traverse in the left subtree
      treePathsSumUtil(root.right, currPath, allPath);  // traverse in the right
subtree

      currPath.remove(currPath.size() - 1);     // remove the current element from the
path
}
static int treePathsSum(Node root) {       // store all the root to leaf path in allPath
      ArrayList<ArrayList<String> > allPath = new ArrayList<>();
      ArrayList<String> v = new ArrayList<>();
      treePathsSumUtil(root, v, allPath);

      int s = 0;     // store the sum
      for (ArrayList<String> pathNumber : allPath) {
      String k = "";
      for (String x : pathNumber)      // join the pathNumbers to convert them into the
number to calculate sum
            k += x;
      s += Integer.parseInt(k);    //data converting
      }
      return s;
}
public static void main(String[] args) {     //main program created
      Node root = new Node(1);
      root.left = new Node(2);
      root.right = new Node(3);
      System.out.println("Sum of all paths is: " + treePathsSum(root));    //calling and
printing the value
  }
}
```

Program: Evaluate the reverse polish notation

```
package com.ishwarchavan;
import java.util.Stack;
```

```
public class EvaluateRevPolNot {     //class created
          public static int evalRPN(String[] tokens) {    //function created
              Stack<Integer> stack = new Stack<>();
              int a,b;    //integer variables

              for(String s: tokens) {    //create new String object in the Stack with
the value "tokens" assign to the variable s.
                  if(s.equals("+"))
                     stack.push(stack.pop()+stack.pop());  //if condition true then
execute it

                  else if(s.equals("-")) {    //else if condition true then execute
below statement
                         a = stack.pop();
                         b = stack.pop();
                         stack.push(b-a);
                     }
                  else if(s.equals("*")) {     //if true then execute below statement
                     stack.push(stack.pop() * stack.pop());
                  }
                  else if(s.equals("/")) {     //if true then execute below statement
                         a = stack.pop();
                         b = stack.pop();
                         stack.push(b/a);
                     }
                  else         //if not satisfied the condition then execute below
statement
                     stack.push(Integer.parseInt(s));
              }

              return stack.pop();     //return the value
          }

public static void main(String[] args) {     //main program created
      String[] tokens = {"10","1","+","3","*"};   //string stored in tokens variable
      System.out.println(evalRPN( tokens));    //calling and printing the value
  }
}
```