```
Script started on 2021-03-06 19:57:18-0600
l_ladios@ares:~$ cat PostFixNotation.into
cat: PostFixNotation.into: No such file or directory
l_ladios@ares:~$ cat PostFixNotation.info
 /**************************************************************************

     *                                                                      *

     *  NAME:  Leia Ladios                        CLASS:  CSC121-W01        *

     *                                                                      *

     *  Assignment: Lab P-6.35                    Level:   2                *

     *                                                                      *

     *  Description:                                                        *

     *                                                                      *

        P-6.35: Implement a program that can input an expression

                        of postfix notation and output its value.

     *                                                                      *

    **************************************************************************/
```

```
l_ladios@ares:~$ cat Stack.java

/**
 * Stack ADT from the book
 * @param <T>
 */
public interface Stack<T> {

   /**
    * Returns number of elements in the stack
    * @return number of elements in the stack
    */
   int size();

   /**
    * Tests if the stack is empty
    * @return true if the stack is empty
    *         false if not empty
    */
   boolean isEmpty();

   /**
    * Inserts an element at the top of the stacks
    * @param t element to be inserted
    */
   void push(T e);

   /**
    * Returns the element at top of the stack
    * but does not remove it
    * @return top element in the stack
    */
   T top();

   /**
    * Removes and returns top element from the stack
    * @return element removed(or null if stack is empty)
    */
   T pop();

}


l_ladios@ares:~$ javac Stack.java
l_ladios@ares:~$ cat ArrayStack.java

import java.lang.reflect.Array;
import java.util.ArrayList;

public class ArrayStack<T> implements Stack<T>{
      public static final int CAPACITY = 50;
      private T[] data;
      private int t = -1;  // empty stack

      public ArrayStack() {
         this(CAPACITY);
      }

      @SuppressWarnings("unchecked")
      public ArrayStack(int capacity) {
         data = (T[])new Object[capacity];
      }

      @Override
      public int size() {
         return t + 1;
      }

      @Override
      public boolean isEmpty() {
         if(t == -1) {
            return true;
         }
         return false;
```

```
}

@Override
public void push(T e) {
   if(size() == data.length) {
      System.out.println("Stack Full");
   }


   data[++t] = e;
}

@Override
public T top() {
   if(isEmpty()) {
      System.out.println("Empty Stack");
      return null;
   }
   return data[t];
}

@Override
public T pop() {
  if(isEmpty()) {
     return null;
  }
  T toRemove = data[t];
  data[t] = null; //dereference
  t--;

  return toRemove;
}

public T evaluate(String pst) {

      for(int i = 0; i < pst.length(); i++) {

         Character curr = pst.charAt(i);

         if(Character.isDigit(curr)) {
            Integer j = new Integer(Character.getNumericValue(curr));
            this.push((T)j);
         }

         if((curr == '%' || curr == '/' || curr == '*'
           || curr == '+' || curr == '-') && size() >= 2) {

            T a = this.pop();
            T b = this.pop();

            if(curr == '%') {
               Integer mod = (Integer)b % (Integer)a;
                  // must cast to do operations
               this.push((T)mod);
               }
               else if (curr == '/') {
                  Integer div = (Integer)b / (Integer)a;
                     // must cast to do operations
                  this.push((T)div);

               }
               else if(curr == '*') {
                  Integer mult = (Integer)b * (Integer)a;
                     // must cast to do operations
                  this.push((T)mult);

               }
               else if(curr == '+') {
                  Integer add = (Integer)b + (Integer)a;
                     // must cast to do operations
                  this.push((T)add);
               }
               else if(curr == '-') {
                  Integer sub = (Integer)b - (Integer)a;
                     // must cast to do operations
                  this.push((T)sub);
                                 }

            }
            else if(i == pst.length() -1) {
               System.out.println("Not enough arguments. Please try again.");
               return null;
            }


         }
      return this.pop();
   }


}
l_ladios@ares:~$ javac ArrayStack.java
Note: ArrayStack.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
l_ladios@ares:~$ cat PostFixMain.java

import java.util.Scanner;
public class PostFixMain {

   public static void main(String[] args) {

      boolean done = false;
      Scanner in = new Scanner(System.in);


      while(!done) {
         System.out.println("Would you like to enter a postfix expression? (Y/N)");
```

```
         String answer = in.next();
         in.nextLine();

         if(answer.toLowerCase().equals("y")) {
            System.out.print("Please enter a postfix expression: ");
            String postFix = in.nextLine();
            System.out.println();

            ArrayStack<Integer> arrStack = new ArrayStack<Integer>(postFix.length()
            System.out.println("Result: " + arrStack.evaluate(postFix) + "\n");
         }
         else {
            done = true;
         }

      }

   }

}


l_ladios@ares:~$ javac PostFixMain.java
l_ladios@ares:~$ java PostFixMain
Would you like to enter a postfix expression? (Y/N)
y
Please enter a postfix expression: 56*46+-

Result: 20

Would you like to enter a postfix expression? (Y/N)
Y
Please enter a postfix expression: 93/67*+

Result: 45

Would you like to enter a postfix expression? (Y/N)
Y
Please enter a postfix expression: 68*45+-67*+

Result: 81

Would you like to enter a postfix expression? (Y/N)
n
l_ladios@ares:~$ exit
exit

Script done on 2021-03-06 20:00:45-0600
```