class PostfixCalculator:

// Function to evaluate a postfix expression

function evaluatePostfix(postfixExpression):

Create an empty stack

for each character in postfixExpression:

if the character is a digit:

Convert the character to an integer and push it onto the stack

else:

if there are fewer than 2 operands in the stack:

Print "Error: Invalid postfix expression (not enough operands)"

Return null

Pop two operands from the stack: operand2 and operand1

if the character is '+':

Push (operand1 + operand2) onto the stack

else if the character is '-':

Push (operand1 - operand2) onto the stack

else if the character is '\*':

Push (operand1 \* operand2) onto the stack

else if the character is '/':

if operand2 is 0:

Print "Error: Division by zero"

Return null

Push (operand1 / operand2) onto the stack

else if the character is '%':

if operand2 is 0:

Print "Error: Division by zero"

Return null

Push (operand1 % operand2) onto the stack

else:

Print "Error: Unknown operator"

Return null

if there is more than one item left in the stack:

Print "Error: Invalid postfix expression (incorrect final stack size)"

Return null

Return the last item in the stack

// Function to evaluate postfix expressions from a file

function evaluateFromFile(filename):

Try to open the file specified by filename

for each line in the file:

Read the postfix expression from the line

Evaluate the expression using evaluatePostfix

if the result is not null:

Print the result

else:

Print "Evaluation failed for expression: " + expression

If an error occurs while opening the file:

Print "Error: Unable to read file"

class Main:

// Main function to test the PostfixCalculator class

function main:

Create a new PostfixCalculator object

// Test evaluatePostfix with several expressions

Print the result of evaluating "23\*4+" // (2 \* 3) + 4 = 10

Print the result of evaluating "55+9\*" // (5 + 5) \* 9 = 90

Print the result of evaluating "93/8+" // (9 / 3) + 8 = 11

Print the result of evaluating "92%4+" // (9 % 2) + 4 = 5

Print the result of evaluating "82/-" // Invalid (not enough operands for '-')

Print the result of evaluating "30/8+" // Error: Division by 0

// Test evaluateFromFile with expressions from "expressions.txt"

Call evaluateFromFile on PostfixCalculator with "expressions.txt"