* Assignment No-10 *

Aim:-

Implement C++ program for expression conversion as infix to postfix and its evolution using stack based on given conditions.

1. operands & operators, both must be single characters.
2. Input postfix expression must be in a desired format.
3. only +, -, * and / operator are expected

Hardware and Software required:

lab Computer, C++ complier, Eclipse for C++.

Prerequisities:-

Basic skills of C++ programming language, C++ stack and polish notations

Theory:-

Stack:

A stack is similar to read life stack or a pile of things that are stack one above another. Stack follows. Ifo order i.e. last-in-first-out

Polish Notation:-

polish notation is a way of expressing arithmatic expression that avoid the use of brackets to define priorities for evaluation of operator. polish Notation was devised by polish philosopher & mathematician Jan Lukasiewicz for use in symbolic logic. In this notation the operator

processed their arguments. So that the Infix Notation expression.

(3+5) * (7-2)

could be written as

* +35-72 (prefix)

The reversed form, Reverse polish Notation (RPN), has however been found more convient from a computational point of view. In this notation the above expression would be

35 + 72 - * (postfix).

Infix Expression:

The Expression of the form a op b. when an operator is in between every pair of operands.

postfix expression: The expression of form a b op. when an operator is followed by every pair of operands.

Prefix expression:

The expression of form op a b,

Algorithm:

Step 1: start.

Step 2: - Include required header file of C++.

Step 3: scan expression E from left to right, character by characters, till character is #

ch = get-next_token(E).

Step 4: while (ch != #)

If (ch = ')') then ch=pop()

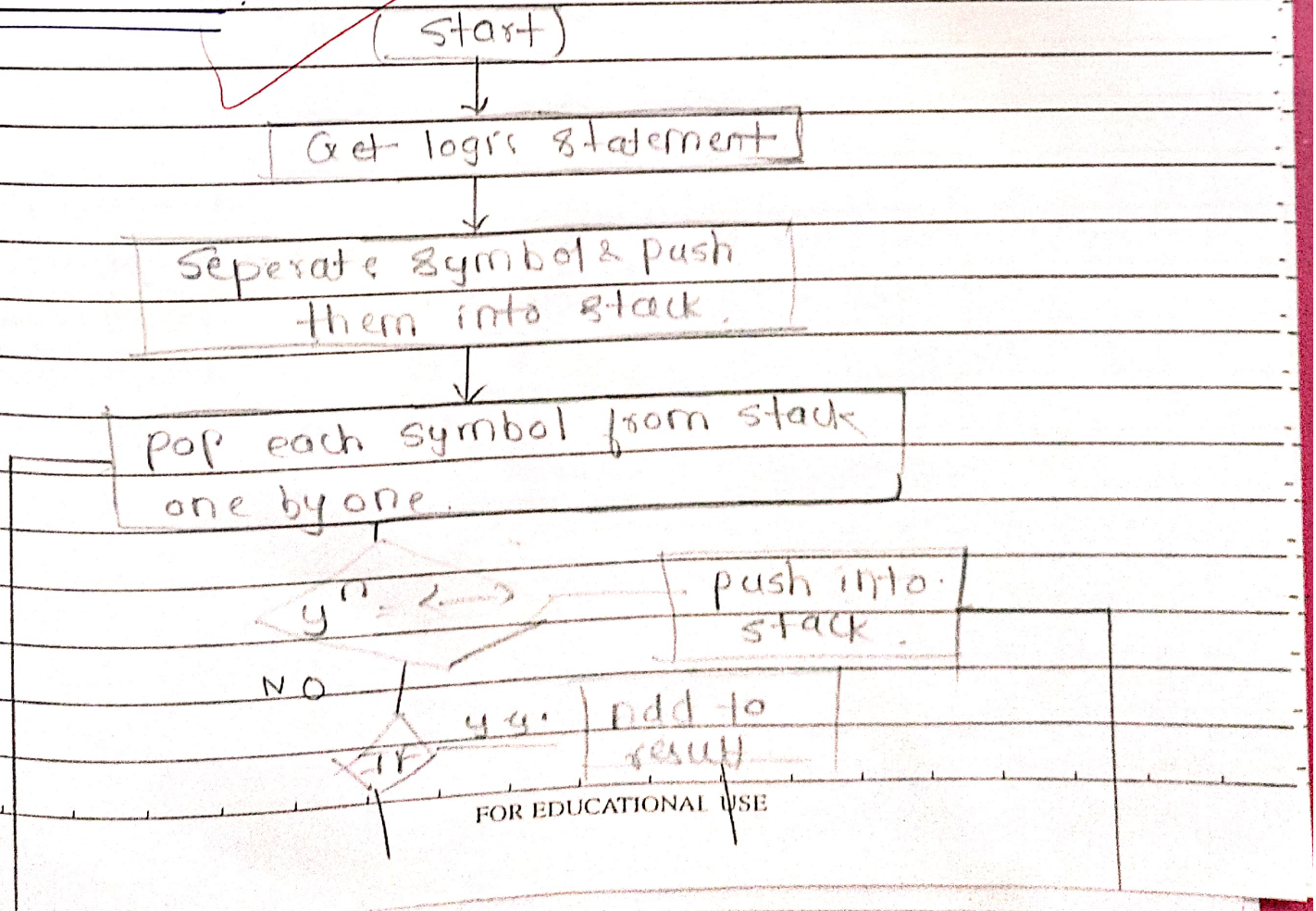
while (ch != '(')

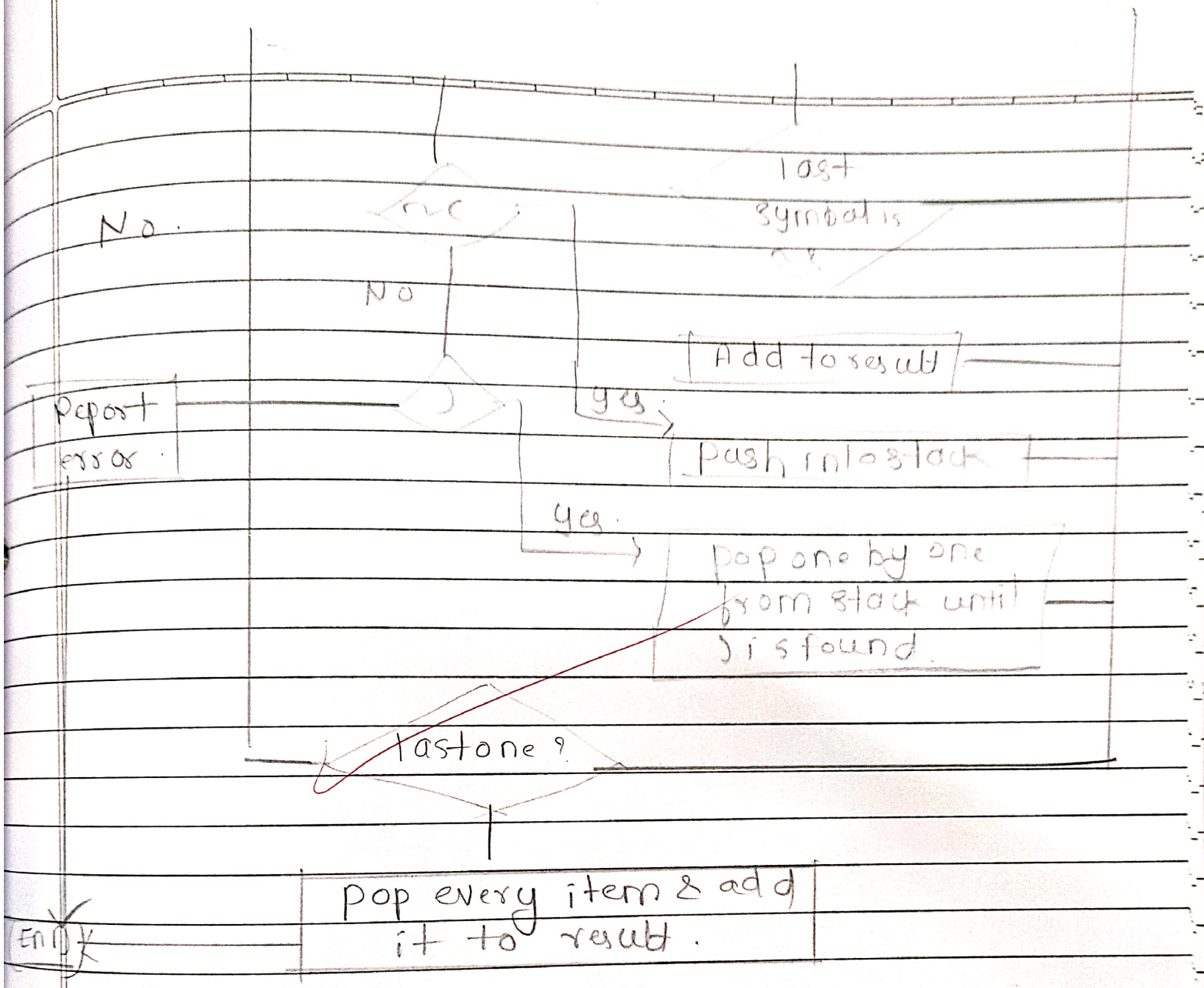
display ch
ch = pop()
end while.
If (ch = operand) display the same.
If (ch = operator) then
If (ICP > ISP) then push ch
else
while (ICP <= ISP)
pop the operator & display it.
end while.
ch = get_next_token(E)
End while.

Step 5: If (ch = #) then while (! Empty stack()) pop & display.

Step 6: stop.

* Flowchart :-





Conclusion :-

Thus we have successfully implemented the Conversion of infix expression to postfix expression and its evaluation using various operation on stack.