10) Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions i. Operands and operator, both must be single character. ii. Input Postfix expression must be in a desired format. iii. Only '+', '-', '\*' and '/ ' operators are expected

***CODE :-***

#include <iostream>

#include <stack>

#include <cctype>

#include <cstdlib>

using namespace std;

bool isOperand(char c)

{

return isalpha(c);

}

int precedence(char op)

{

if (op == '+' || op == '-') return 1;

if (op == '\*' || op == '/') return 2;

return 0;

}

int applyOperator(int a, int b, char op)

{

switch(op)

{

case '+': return a + b;

case '-': return a - b;

case '\*': return a \* b;

case '/': return a / b;

default: return 0;

}

}

string infixToPostfix(string infix)

{

stack<char> s;

string postfix = "";

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

{

char c = infix[i];

if (isOperand(c))

{

postfix += c;

}

else if (c == '(')

{

s.push(c);

}

else if (c == ')')

{

while (!s.empty() && s.top() != '(')

{

postfix += s.top();

s.pop();

}

s.pop(); // Pop '('

}

else if (c == '+' || c == '-' || c == '\*' || c == '/')

{

while (!s.empty() && precedence(s.top()) >= precedence(c))

{

postfix += s.top();

s.pop();

}

s.push(c);

}

}

while (!s.empty())

{

postfix += s.top();

s.pop();

}

return postfix;

}

int evaluatePostfix(string postfix)

{

stack<int> s;

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

{

char c = postfix[i];

if (isOperand(c))

{

cout << "Enter value for operand " << c << ": ";

int value;

cin >> value;

s.push(value);

}

else if (c == '+' || c == '-' || c == '\*' || c == '/')

{

int b = s.top(); s.pop();

int a = s.top(); s.pop();

int result = applyOperator(a, b, c);

s.push(result);

}

}

return s.top();

}

int main()

{

string infix;

cout << "Enter infix expression (single character operands and operators only): ";

cin >> infix;

string postfix = infixToPostfix(infix);

cout << "Postfix expression: " << postfix << endl;

int result = evaluatePostfix(postfix);

cout << "Result of evaluation: " << result << endl;

return 0;

}

***OUTPUT :-***

