Name - Manaswi Santosh Kulkarni

Roll No - 47

PRN - F23112054

Q batch Comp-2

Group D-27

Implement C++ program for expression conversion as infix to postfix and its evaluation

using stack based on given conditions:

1. Operands and operator, both must be single character.

2. Input Postfix expression must be in a desired format.

3. Only '+', '-', '\*' and '/ ' operators are expected.

#include<iostream>

#define MAX\_ARR\_SIZE 30

using namespace std;

int VAR\_VALS[30] = {0};

int VAR\_COUNT = 0;

bool IS\_VAL\_SET = false;

class char\_stack{

char list[MAX\_ARR\_SIZE];

public:

int top;

char\_stack(){

top = -1;

}

bool isempty(){ return (top <=-1) ? true : false; }

bool isfull(){ return (top==MAX\_ARR\_SIZE - 1) ? true : false; }

void push(char elmt){

if(not isfull()){

top++;

list[top] = elmt;

}

}

char peek() { return (top !=1) ? list[top] : ' ';}

char pop(){

if(not isempty()){

top--;

return list[top+1];

}

}

};

class stack{

int list[MAX\_ARR\_SIZE];

public:

int top;

stack(){

top = -1;

}

bool isempty(){

return (top <=-1) ? true : false;

}

bool isfull(){

return (top == MAX\_ARR\_SIZE -1) ? true : false;

}

void push(int elmt){

if(not isfull()){

top++;

list[top]=elmt;

}

}

int pop(){

if(not isempty()){

top--;

return list[top + 1];

}

}

void display(){

for(int i=0;i<top + 1;i++){

cout<<list[i]<<" ";

}

cout<<endl;

}

};

int priority(char opp){

if(opp == '+' || opp == '-'){

return 1;

}else if(opp == '\*' || opp == '/'){

return 2;

}else{

return 0;

}

}

string infixToPostfix(string exp){

string postfix;

char ch;

char\_stack opps;

for(int i=0; i< exp.size(); i++){

ch = exp[i];

if(ch=='('){

opps.push('(');

}else if(ch == '+' || ch == '^' || ch == '-' || ch == '\*'|| ch =='/'){

while(!opps.isempty()&& priority(ch)<=priority(opps.peek())){

postfix += opps.pop();

}

opps.push(ch);

}else if(ch == ')'){

while(opps.peek() != '('){

postfix += opps.pop();

}

opps.pop();

}else{

postfix +=ch;

}

}

while(!opps.isempty()){

postfix+=opps.pop();

}

return postfix;

}

void setVariables(int count = 0){

int varnum;

if(count == 0){

cout<<"Enter Number of Variables in Expression: ";

cin>> varnum;

}else{

varnum = count;

}

VAR\_COUNT = varnum+ 64;

for(int i=0;i<varnum;i++){

cout<<"Enter Value for '"<<(char)(65+i)<<"' : ";

cin>>VAR\_VALS[i];

VAR\_VALS[i+32] = VAR\_VALS[i];

}

IS\_VAL\_SET = true;

}

int evalPostfix(string exp){

int max\_var = 0;

for(int i=0;i< exp.size();i++){

if(exp[i]>65 && exp[i] <=90){

if(exp[i]>max\_var){

max\_var = exp[i];

}

}else if(exp[i]>97&& exp[i] <=122){

if((exp[i]-32) > max\_var){

max\_var = (exp[i]-32);

}

}

}

if(!IS\_VAL\_SET || max\_var > VAR\_COUNT){

setVariables(max\_var-64);

}

int len = exp.size();

int val1,val2,ans,temp;

stack values;

for(int i=0;i<len;i++){

switch(exp[i]){

case '+':

val2=values.pop();

val1=values.pop();

temp = val1+val2;

values.push(temp);

break;

case '\*':

val2 = values.pop();

val1 = values.pop();

temp = val1\*val2;

values.push(temp);

break;

case '-':

val2 = values.pop();

val1 = values.pop();

temp = val1-val2;

values.push(temp);

break;

case '/':

val2 = values.pop();

val1 = values.pop();

temp = val1/val2;

values.push(temp);

break;

case ' ':

break;

default:

values.push(VAR\_VALS[int (exp[i] - 65)]);

break;

}

}

return values.pop();

}

int main(){

int ch;

bool while\_ctrl = true;

string temp,LastInfixExp;

int ans;

while(while\_ctrl){

cout<<"MENU\n1. Infix to Postfix \n2. Set Variable Values \n3. Postfix Calculator \n4. Exit"<<endl;

cout<<"Enter Your Choice: ";

cin>>ch;

switch(ch){

case 1:

cout<<"Enter a Infix Expression: ";

cin>>temp;

LastInfixExp = infixToPostfix(temp);

cout<<"Postfix Expression is: "<<LastInfixExp <<endl;

break;

case 2:

setVariables();

cout<<"Variables set successfully"<<endl;

break;

case 3:

if(LastInfixExp.size() !=0){

cout<<"Enter Postfix Expression(Enter 0 to use last converted expression):";

cin>>temp;

if(temp=="0"){

ans=evalPostfix(temp);

}else{

ans=evalPostfix(LastInfixExp);

}

cout<<"Value of Expression is: "<<ans<<endl;

}else{

cout<<"Enter Postfix Expression:";

cin>>temp;

ans = evalPostfix(temp);

cout<<"Value of expression is: "<< ans <<endl;

}

break;

case 4:

cout<<"Thank You!"<<endl;

while\_ctrl = false;

break;

default:

cout<<"Enter a valid Choice!"<<endl;

break;

}

}

OUTPUT

MENU

1. Infix to Postfix

2. Set Variable Values

3. Postfix Calculator

4. Exit

Enter Your Choice: 1

Enter a Infix Expression: 22

Postfix Expression is: 22

MENU

1. Infix to Postfix

2. Set Variable Values

3. Postfix Calculator

4. Exit

Enter Your Choice: 2

Enter Number of Variables in Expression: 2

Enter Value for 'A' : 1

Enter Value for 'B' : 3

Variables set successfully

MENU

1. Infix to Postfix

2. Set Variable Values

3. Postfix Calculator

4. Exit

Enter Your Choice: 3

Enter Postfix Expression(Enter 0 to use last converted expression):0

Value of Expression is: 0