#include <iostream>

#include <stack>

#include <string>

#include <stdlib.h>

using namespace std;

class Tree

{

public:

char oper;

Tree\* right;

Tree\* left;

Tree()

{

right=NULL;

left=NULL;

}

friend class list;

};

class list

{

public:

Tree \*root;

list()

{

root=NULL;

}

void create();

void in(Tree\*);

void post(Tree\*);

void pre(Tree\*);

void in1()

{

Tree \*temp = root;

in(temp);

}

void post1()

{

Tree \*temp = root;

post(temp);

}

void pre1()

{

Tree \*temp = root;

pre(temp);

}

int priority(char x)

{

if(x=='+' || x=='-')

{

return 1;

}

else if(x=='/' || x=='\*')

{

return 2;

}

else

return 3;

}

};

void list::create()

{

string input;

stack<Tree\*> s;

stack<Tree\*> s1;

cout << "\nPlease enter an expression: " << endl;

cin >> input;

int p,T;

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

{

p=priority(input[i]);

if(p==3)

{

Tree \*temp = new Tree;

temp->oper = input[i];

s.push(temp);

}

else

{

Tree \*temp = new Tree;

temp->oper = input[i];

if(s1.empty())

{

s1.push(temp);

}

else

{

Tree\* t = s1.top();

T = priority(t->oper);

while(p<=T && (!s.empty() && !s1.empty()))

{

Tree \*op = s1.top();

s1.pop();

Tree \*x1 = s.top();

s.pop();

Tree \*x2 = s.top();

s.pop();

op->right = x1;

op->left = x2;

root = op;

s.push(op);

}

s1.push(temp);

}

}

}

while(!s.empty() && !s1.empty())

{

Tree \*op = s1.top();

s1.pop();

Tree \*x1 = s.top();

s.pop();

Tree \*x2 = s.top();

s.pop();

op->right = x1;

op->left = x2;

root = op;

s.push(op);

}

}

void list::in(Tree \*temp)

{

stack<Tree \*> s3;

while(1)

{

while(temp)

{

s3.push(temp);

temp = temp->left;

}

if(s3.empty())

{

return ;

}

else

{

temp = s3.top();

s3.pop();

cout<<temp->oper;

temp = temp->right;

}

}

}

void list::in(Tree \*temp)

{

if(temp!=NULL)

{

in(temp->left);

cout<<temp->oper;

in(temp->right);

}

}

void list::post(Tree \*temp)

{

if(temp!=NULL)

{

post(temp->left);

post(temp->right);

cout<<temp->oper;

}

}

void list::pre(Tree \*temp)

{

if(temp!=NULL)

{

cout<<temp->oper;

pre(temp->left);

pre(temp->right);

}

}

int main()

{

list l;

int ch;

l.create();

cout<<endl;

while(1)

{

cout<<"\n1.Preorder: \n";

cout<<"\n2.Inrder: \n";

cout<<"\n3.Postorder: \n";

cout<<"\n4.Exit \n";

cout<<"\nEnter the Choice: ";

cin>>ch;

switch(ch)

{

case 1:

l.pre1();

break;

case 2:

l.in1();

break;

case 3:

l.post1();

break;

case 4:

exit(1);

break;

default : cout<<"\nWrong choice\n";

break;

}

}

return 0;

}