Expression Tree

#include<iostream.h>

#include<conio.h>

#include<process.h>

#include<stdio.h>

#define max 30

struct node

{ char data;

node \*left, \*right;

} \*root=NULL, \*temp, \*t1, \*t2;

char exp[max],post[max],s[max];

char op1;

int te,tp,top,i;

node \*q[max];

int priority(char op)

{ if(op=='^')

return 3;

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

return 2;

else if(op=='+'||op=='-')

return 1;

else

return 0;

}

void push(char item)

{ if(top==(max-1))

{ cout<<"\nStack is full.";

}

else

{ top++;

s[top]=item;

}

}

char pop(void)

{ char item;

if(top==-1)

{ cout<<"\nStack is empty.";

return -1;

}

else

{ item=s[top];

top--;

return item;

}

}

void postfix(void)

{ tp=0;

push('(');

cout<<endl;

for(i=0;i<te;i++)

{ if(exp[i]=='+'||exp[i]=='-'||exp[i]=='/'||exp[i]=='\*'||exp[i]=='^')

{ up1:

op1=pop();

if(op1=='(')

{ push(op1);

push(exp[i]);

}

else

{ if(priority(op1)>=priority(exp[i]))

{ cout<<op1;

post[tp]=op1;

tp++;

goto up1;

}

else

{ push(op1);

push(exp[i]);

}

}

}

else if(exp[i]=='(')

{ push(exp[i]);

}

else if(exp[i]==')')

{ up2:

op1=pop();

if(op1!='(')

{ cout<<op1;

post[tp]=op1;

tp++;

goto up2;

}

}

else

{ cout<<exp[i];

post[tp]=exp[i];

tp++;

}

}

cout<<"\nPostfix Expression created.\n";

}

void tree(void)

{ top=-1;

for(i=0;i<tp;i++)

{ temp=new node;

temp->data=post[i];

temp->left=NULL;

temp->right=NULL;

if(post[i]=='+'||post[i]=='-'||post[i]=='/'||post[i]=='\*'||post[i]=='^')

{ t2=q[top];

top--;

t1=q[top];

top--;

temp->right=t2;

temp->left=t1;

top++;

q[top]=temp;

}

else

{ top++;

q[top]=temp;

}

}

root=q[top];

}

void display(node \*ptr)

{ if(ptr!=NULL)

{ cout<<ptr->data;

display(ptr->left);

display(ptr->right);

}

}

void main()

{ te=0; tp=0;

top=-1;

clrscr();

cout<<"Enter the expression.\n";

gets(exp);

for(i=0;exp[i]!='\0';i++);

exp[i]=')';

te=i+1;

postfix();

tree();

cout<<"\nTree (Preorder traversal):\n";

display(root);

getch();

}

