**INFIX TO POSTFIX CONVERSION USING STACK**

#include<iostream.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

constint size=50;

char infix[size],postfix[size],Stack[size];

int top=-1;

int precedence(char ch);

char Pop();

charTopelement();

void Push(char ch);

int braces(char\*);

int main()

{ charele,elem,st[2];

intprep,pre,popped,j=0,chk=0;

strcpy(postfix," ");

system("cls");

cout<<"ASSUMPTION:The infix expression contains singlr letter variables\n\t"

<<"\t and single digit constants only. \n\n";

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

gets(infix);

chk=braces(infix);

if(chk!=0)

{ cout<<"Unbalanced no. of braces.\nExtra";

cout<<(chk==1?"right braces":"left braces")<<endl;

system("pause"); exit(1);

}

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

{ if(infix[i]!='('&&infix[i]!=')'&& infix[i]!='^'&&infix[i]!='\*'

&&infix[i]!='/'&&infix[i]!='+'&&infix[i]!='-')

postfix[j++]=infix[i];

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

{ elem=infix[i];

Push(elem);

}

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

{ while( (popped=Pop()) !=')' )

{ postfix[j++]=popped;}

}

else

{ elem=infix[i];

pre=precedence(elem);

ele=Topelement();

prep=precedence(ele);

if(pre>prep) Push(elem);

else

{ while(prep>=pre)

{ if(ele=='#') break;

popped=Pop();

ele=Topelement();

postfix[j++]=popped;

prep=precedence(ele);

}

Push(elem);

}

}

}

while( (popped=Pop())!='#')

postfix[j++]=popped;

postfix[j]='\0';

cout<<"\n Post fix :"<<postfix<<endl; system("pause");

return 0;

}

int precedence(char ch)

{ switch(ch)

{ case '^':return 5;

case '/':return 4;

case '\*':return 4;

case '+':return 3;

case '-':return 3;

default :return 0;

}

}

char Pop()

{ char ret;

if(top!=-1)

{ ret=Stack[top];

top--;

return ret;

}

else

return'#';

}

charTopelement()

{ charch;

if (top!=-1) ch=Stack[top];

elsech='#';

returnch;

}

void Push(char ch)

{ if (top!=size-1)

{ top++;

Stack[top]=ch;

}

}

int braces(char\*s)

{ intleftbr,rightbr;

leftbr=rightbr=0;

for(inti=0;s[i];i++)

{ if(s[i]=='(') leftbr++;

else if (s[i]==')') rightbr++;

}

if(leftbr==rightbr)return 0;

else if(leftbr<rightbr) return 1;

else return -1;

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

