//INFIX TO POSTFIX

#include <stdio.h>

#include <conio.h>

#include <ctype.h>

#include <string.h>

#include <stdlib.h>

#define MAX 100

char st[MAX];

int top=-1;

void push(char st[],char);

char pop(char st[]);

void InfixToPostfix(char source[],char target[]);

int getPriority(char);

int main()

{

char infix[100],postfix[100];

printf("\n Enter any infix expression\n");

gets(infix);

strcpy(postfix,"");

InfixToPostfix(infix,postfix);

printf("\nRequired postfix expression is \n");

puts(postfix);

getch();

return 0;

}

void InfixToPostfix(char source[],char target[])

{

int i=0,j=0;

char temp;

strcpy(target,"");

while (source[i]!='\0')

{

if (source[i]=='(')

{

push(st,source[i]);

i++;

}

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

{

while ((top!=-1)&&(st[top]!='('))

{

target[j]=pop(st);

j++;

}

if(top==-1)

{

printf("\nIncorrect expression\n");

exit(1);

}

temp=pop(st);

i++;

}

else if(isdigit(source[i])||isalpha(source[i]))

{

target[j]=source[i];

j++;

i++;

}

else if (source[i]=='+'||source[i]=='-'||source[i]=='\*'||source[i]=='/'||source[i]=='%')

{

while((top!=-1)&&(st[top]!='(')&&(getPriority(st[top])>getPriority(source[i])))

{

target[j]=pop(st);

j++;

}

push(st,source[i]);

i++;

}

else

{

printf("\nIncorrect element in expression\n");

exit(1);

}

}

while((top!=-1)&&(st[top]!='('))

{

target[j]=pop(st);

j++;

}

target[j]='\0';

}

int getPriority(char op)

{

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

return 1;

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

return 0;

}

void push(char st[],char val)

{

if (top==MAX-1)

printf("\nStack Overflow");

else

{

top++;

st[top]=val;

}

}

char pop(char st[])

{

char val=' ';

if(top==-1)

printf("\nStack underflow");

else

{

val=st[top];

top--;

}

return val;

}

OUTPUT-



