**Lab 2 : InfixToPostfix Expression conversion using Stack operations.**

**Code:**

#include<stdio.h>

#include<string.h>

#include<conio.h>

#include<stdlib.h>

#define SIZE 100

char st[SIZE];

int top = -1;

void push(char st[], char);

char pop(char st[]);

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

int getPriority(char);

int main(){

char infix[100], postfix[100];

printf("Enter: \n");

gets(infix);

strcpy(postfix,"");

InfixToPostfix(infix,postfix);

printf("Required : \n");

puts(postfix);

getch();

return 0;

}

void InfixToPostfix(char source[], char dest[]){

int i=0, j=0;

char temp;

strcpy(dest,"");

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

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

push(st, source[i]);

i++;

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

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

dest[j]=pop(st);

j++;

}

if(top==-1){

printf("INCORRECT EXPRESSION");

exit(0);

}

temp = pop(st);

i++;

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

dest[j]=source[i];

j++;

i++;

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

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

dest[j]=pop(st);

j++;

}

push(st, source[i]);

i++;

}else{

printf("INCORRECT ELEMENT");

exit(0);

}

}

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

dest[j]=pop(st);

j++;

}

dest[j]='\0';

}

int getPriority(char c){

if(c=='/' || c=='\*' || c=='%') return 1;

else if(c=='+'||c=='-') return 0;

}

void push(char st[], char val){

if (top==SIZE-1){

printf("STACK OVERFLOW");

}else{

st[++top]=val;

}

}

char pop(char st[]){

char val=' ';

if(top==-1){

printf("STACK UNDERFLOW");

}else{

val=st[top];

top--;

}

return val;

}

**Output:**

