**Stack1.h**

#define STACKSIZE 10

typedef struct stack{

int s1[STACKSIZE];

int top;

}stack;

void createstack(stack\*s)

{

s->top=-1;

}

int isfull(stack\*s)

{

if(s->top==(STACKSIZE-1))

return 1;

else

return 0;

}

int isempty(stack\*s)

{

if (s->top==-1)

return 1;

else

return 0;

}

void push(stack\*s,int n)

{

s->top++;

s->s1[s->top]=n;

}

int pop(stack\*s)

{

int ele;

ele=s->s1[s->top];

s->top-- ;

return ele;

}

int peek (stack\*s)

{

return (s->s1[s->top]);

}

void display(stack\*s)

{

int i;

for(i=s->top ; i>=0;i--)

{

printf("%d\n",s->s1[i]);

}

}

**INFIX TO POSTFIX**

#include<conio.h>

#include<stdio.h>

#include<stack1.h>

#define SIZE 25

int getprecedence(char ch)

{

if(ch=='^')

return 4;

else if (ch=='\*'||ch=='/'||ch=='%')

return 3;

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

return 2;

else

return 0 ;

}

void infix\_to\_postfix(char infix[],char postfix[])

{

char ch;

stack s;

int i,j=0;

createstack(&s);

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

{

ch=infix[i];

if(ch=='(')

push(&s,ch);

else if(ch>='0' && ch<='9')

postfix[j++]=ch;

else if(ch=='^'||ch=='/'||ch=='+'||ch=='-'||ch=='\*'||ch=='%')

{

while(!isempty(&s)&&getprecedence(peek(&s))>=getprecedence(ch))

{

postfix[j]=ch;

j++ ;

}

push(&s,ch) ;

}

else if (ch==')')

{

while(peek(&s)!='(')

{

postfix[j++]=pop(&s);

}

pop(&s);

}

}

while(!isempty(&s))

{

postfix[j++]=pop(&s);

}

postfix[j]='\0';

}

void main()

{

char infixexp[SIZE],postfixexp[SIZE];

clrscr();

printf("Enter the infix expression : ");

scanf("%s",infixexp);

infix\_to\_postfix (infixexp,postfixexp);

printf("Postfix of Expression is:%s",postfixexp);

getch();

}