**4.Write a C program for conversion of expression given, from infix to prefix.**

**ALGORITHM**

Step 1: Scan the infix expression from left to right

Step 2: If the scanned character is an operand, append it with final infix to prefix string

Step 3: Else, if the precedence order of the scanned operator is greater than the precedence order of the operator in the stack

Step 4: Else, pop all the operators from the stack which are greater than or equal to in precedence than that of the scanned operator. After doing that push the scanned operator to the stack

Step 5: If the scanned character is a ‘(‘ or ‘)’, push it to the stack

Step 6: Repeat the steps until infix expression is scanned

Step 7:Print the output

Step 8: Pop and output from the stack until it is not empty

**PROGRAM**

#include<stdio.h> //standard I/O header file

#include<stdlib.h> //standard library header file

#include<ctype.h>

#include<string.h>

#define max 100 //declaring the size to 5

int top=-1, a[max];

void push(char x) //push function

{

a[++top]=x;

}

char pop() //pop function

{

if(top==-1)

return -1;

else

return a[top--];

}

int prcd(char c) //precedence function

{

if(c==')') //checks the condition and enters the while loop

return 0;

else if(c=='+'||c=='-') //checks the condition and enters the while loop

return 1;

else if(c=='\*'||c=='/') //checks the condition and enters the while loop

return 2;

}

void strrev(char \*exp)

{

char temp[50];

int size=strlen(exp);

temp[size--]='\0';

int i=0;

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

{

temp[size]=exp[i];

i++;

size--;

}

strcpy(exp,temp);

}

void infixtoprefix(char infix[max],char prefix[max]) //infix to prefix function

{

char temp,x; //declaring temporary element

int i=0,j=0;

strrev(infix);

while(infix[i]!='\0') //checks the condition and enters the while loop

{

temp=infix[i];

if(isalnum(temp)) //checks the condition and enters the while loop

{

prefix[j++]=temp;

}

else if(temp==')') //checks the condition and enters the while loop

push(temp);

else if(temp=='(') //checks the condition and enters the while loop

{

while((x=pop())!=')') //checks the condition and enters the while loop

{

prefix[j++]=x;

}

}

else //checks the condition and enters the while loop

{

while(prcd(a[top])>=prcd(temp)) //checks the condition and enters the while loop

{

prefix[j++]=pop();}

push(temp);

}

i++;

}

while(top!= -1) //checks the condition and enters the while loop

prefix[j++]=pop();

prefix[j]='\0';

strrev(prefix);

}

int main() //main fuction

{

char infix[max],prefix[max];

printf("Enter the infix expression\n"); //printing statement

scanf("%s",infix);

printf("The infix expression is %s\n",infix); //printing statement

infixtoprefix(infix, prefix);

printf("The prefix expression is %s\n",prefix); //printing statement

return 0;

}

**OUTPUT**



