S.No: 39	Exp. Name: ***Infix to Prefix Conversion***	Date:2023-06-24

Aim:

Write a C program to convert an Infix expression to Prefix expression.

Source Code:

infixToPrefix.c

```
#define SIZE 50
#include<string.h>
#include <ctype.h>
#include<stdio.h>
char *strrev(char *str)
{
   char c, *front, *back;
   if(!str || !*str)
   {
      return str;
   }
   for(front=str,back=str+strlen(str)-1;front < back;front++,back--)
   {
      c=*front;
      *front=*back;
      *back=c;
   }
   return str;
}
char s[SIZE];
int top = -1;
void push (char elem)
{
   s[++top] = elem;
}
char pop()
{
   return (s[top--]);
}
int pr (char elem)
{
   switch (elem)
   {
      case '#':
      return 0;
      case ')':
      return 1;
      case '+':
      case '-':
      return 2;
      case '*':
      case '/':
      return 3;
   }
}
void main()
```


ID: 224G1A0540

Srinivasa Ramanujan Institute of Technology

```
{
   char infx[50], prfx[50], ch, elem;
   int i = 0, k = 0;
   printf("Enter Infix Expression:");
   scanf ("%s", infx);
   push ('#');
   strrev (infx);
   while ((ch = infx[i++]) != '\0')
   {
      if (ch == ')')
      push (ch);
      else if (isalnum (ch))
      prfx[k++] = ch;
      else if (ch == ')')
      {
         while (s[top] != ')')
         {
            prfx[k++] = pop ();
         }
         elem = pop ();
      }
      else
      {
         while (pr (s[top]) >= pr (ch))
         {
            prfx[k++] = pop ();
         }
         push (ch);
      }
   }
   while (s[top] != '#')
   {
      prfx[k++] = pop ();
   }
   prfx[k] = '\0';
   strrev (prfx);
   strrev (infx);
   printf("Prefix Expression:%s\n", prfx);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter Infix Expression: A+B
Prefix Expression:+AB

Test Case - 2
User Output
Enter Infix Expression: A/B+C/D
Prefix Expression:+/AB/CD

ID: 224G1A0540
2022-2026-CSE-A
Srinivasa Ramanujan Institute of Technology