Aim:- Conversion of Infix Expression to Postfix Expression Using Stack.

Code:-

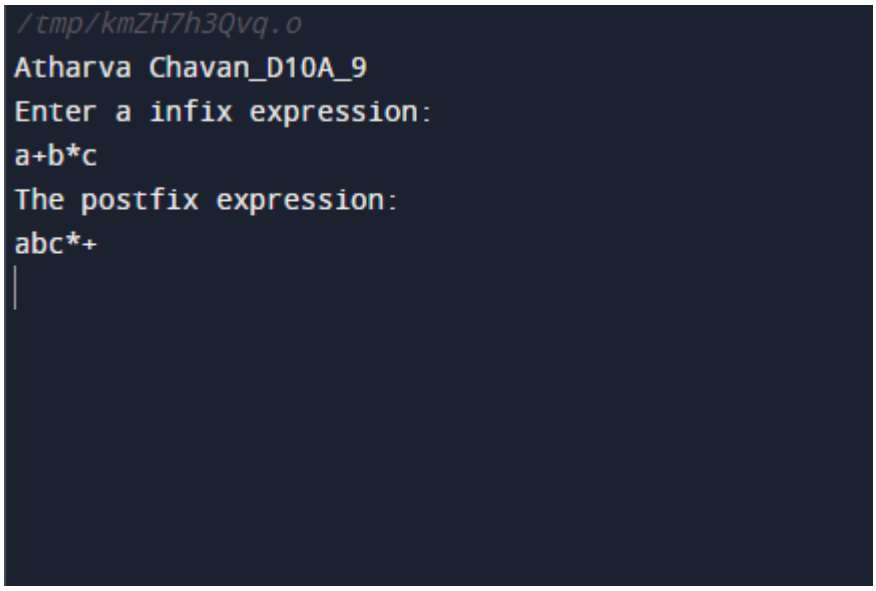
```
#include <stdio.h>

#include <ctype.h>


typedef struct

{

char a[100];

int top;

}stack;

void push (stack *s, char x)

{

if (s->top==99)

printf("Stack is full.");

else

s->a[++s->top]=x;

}

int pop (stack *s)

{

char x;

if(s->top<0)
```

```
printf("Stack is empty");
else
{
x=s->a[s->top--];
return x;
}
}
int precede(char c1,char c2)
{
switch(c1)
{
case '-':
case '+': if(c2=='-'||c2=='+'||c2==')')
return 1;
else
return 0;
case '/':
case '*': if(c2=='(')
return 0;
else
return 1;
case '(':return 0;
}
```

```
}
void conversion(char in[], char post[])
{
stack s;
int i,j=0;
char top;
s.top=-1;
for(i=0;in[i]!='\0';++i)
{
if(isalpha(in[i]))
post[j++]=in[i];
else
{
while(s.top>0&&precede(s.a[top],in[i]))
{
post[j++]=pop(&s);
}
if(in[i]==')')
s.top--;
else
push(&s, in[i]);
}
}
```

```
while(s.top>=0)
{
post[j++]=pop(&s);
post[j]='\0';
}
}
void main()
{
printf("Atharva Chavan_D10A_9\n");
char infix[100], postfix[100];
printf("Enter a infix expression:\n");
gets(infix);
(conversion(infix,postfix));
printf("The postfix expression:\n");
puts(postfix);
}
```

Output:-


/tmp/kmZH7h3Qvq.o
Atharva Chavan_D10A_9
Enter a infix expression:
a+b*c
The postfix expression:
abc*+