EXP NO:05 IMPLEMENTATION OF INFIX TO POSTFIX DATE:

AIM : TO IMPLEMENT INFIX TO POSTFIX ON AN EXPRESSION.

Code:

```
#include <stdio.h>

#include<string.h>
#include<ctype.h>
#define MAX 9
char stack[MAX];
int tos=-1;

void push(char a)
{
if(tos==MAX-1)
{
printf("\nStack Overflow");
return;
}
tos++;
stack[tos]=a;
}

void pop()
{
char a;
if(tos==-1)
{
printf("\nStack Underflow");
return;
}
a=stack[tos];
tos--;
printf("%c",a);
}
int precedence(char a,char b)
{
int precedence;
if(a=='+'&& b=='+' || a=='-' && b=='-')
{
precedence=3;
}

else if(a=='*' || a=='/' || a=='$')
{
precedence=2;
}
else if(a=='+' || a=='-')
```

```
{
precedence=1;
}
else if(a=='(')
{
precedence=0;
}
return precedence;

}
int main()
{

char expr[20];
printf("Enter expr: ");
scanf("%s",expr);
int n= strlen(expr);
for (int i = 0; i < n; i++)
{
if(isdigit(expr[i]))
{
printf("%c",expr[i]);
}
else if (expr[i]=='(')
{
push(expr[i]);
}
else if (expr[i]==')')
{
while(stack[tos]!='(')
{pop();}
tos--;
}
else
{
if (precedence(expr[i],expr[i-1])!=3)
{
if(tos==-1 || precedence(expr[i],expr[i+1])>precedence(stack[tos],stack[tos+1]))
{
if(precedence(expr[i],expr[i+1])==3)
{ push(expr[i]);
push(expr[i+1]);
}
else
push(expr[i]);

}
else
{
```

```
while(precedence(expr[i],expr[i+1])<=precedence(stack[tos],stack[tos+1]) && tos!=-1)
{
if(precedence(expr[i],expr[i+1])==3)
{
pop();
pop();
}
else if (precedence(expr[i],expr[i-1])==3)
{
break;
}
else
pop();
}
push(expr[i]);
}
}
}
}
while(tos!=-1)
{
pop();
}
return 0;
}
```

Output:

```
Enter expr: (12+3)*(88-34)
123+8834-*
```