

## EXPERIMENT NO.3

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>

#define SIZE 100
char stack[SIZE];
int top=-1;
/*define push operation*/
void push(char item)
{
    if(top>=SIZE-1)
    {
        printf("Stack OVERFLOW\n");
    }
    else
    {
        top=top+1;
        stack[top]=item;
    }
}
char pop()
{
    char item;
    if(top<0)
    {
        printf("Stack UNDERFLOW\n");
        getchar();
        exit(1);
    }
    else
    {
        item=stack[top];
        top=top-1;
        return(item);
    }
}
int is_operator(char symbol)
{
    if(symbol=='^' || symbol=='*' || symbol=='/' || symbol=='+' || symbol=='-')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
int is_precedence(char symbol)
{
    if(symbol=='^')
    {
        return(3);
    }
    else if(symbol=='*' || symbol=='/')
    {
        return(2);
    }
    else if(symbol=='+' || symbol=='-')
    {
        return(1);
    }
    else
    {
        return(0);
    }
}
void infixtopostfix(char infix_exp[], char postfix_exp[])
{
    int i,j;
    char item;
    char x;

    push('(');
    strcat(infix_exp,"");
    i=0;
    j=0;
    item=infix_exp[i];
    while(item!='\0')
    {
        if(item=='(')
```

```

push(item);
}
else if(isdigit(item)||isalpha(item))
{
postfix_exp[j]=item;
j++;
}
else if(is_operator(item)==1)
{
x=pop();
while(is_operator(x)==1 && is_precedence(x)>=is_precedence(item))
{
postfix_exp[j]=x;
j++;
x=pop();
}
push(x);
push(item);
}
else if(item=='(')
{
x=pop();
while(x!='(')
{
postfix_exp[j]=x;
j++;
x=pop();
}
}
else
{
printf("Invalid Expression\n");
getchar();
exit(1);
}
i++;
item=infix_exp[i];
}
if(top>0)
{

```

```

R408@ltdnir:~$ gcc Eopt2.c
R408@ltdnir:~$ gcc Eopt2.c
Eopt2.c: In function 'main':
Eopt2.c:133:1: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
133 | gets(infix);
    | ^~~~~
    | fgets
/usr/bin/ld: /tmp/cc4g8PM.o: in function 'main':
Eopt2.c:(.text+0x132): warning: the 'gets' function is dangerous and should not be used.
R408@ltdnir:~$ ./a.out
Enter infix expression
A+B+C*E
Postfix Expression: AB+CC*+
R408@ltdnir:~$

```

```

printf("Invalid expression\n");
getchar();
exit(1);
}
postfix_exp[j]='\0';
}

int main()
{
char infix[SIZE],postfix[SIZE];
printf("Enter infix expression\n");
gets(infix);
infixtopostfix(infix,postfix);
printf("Postfix Expression: ");
puts(postfix);
return 0;
}

```