```c
/*
 Titttle:- CONVERSION OF INFIX TO POSTFIX.
 Name:- TAUSEEF MUSHTAQUE ALI SHAIKH
 Roll no.:- 18CO63
 Class:- S.Y. [CO]
 Date:- 29-07-2019
 DISCRIPTION:- This Program is made to convert a Infix String
 given by the user into Equivalent Postfix String.
 */
#include<stdio.h>
#include<stdbool.h>
#include<unistd.h>
#define MAX 50



struct stack
{
char data[MAX];
int top;
};

bool empty(struct stack *s)
{
    return (s->top==-1)?true:false;
}

void push(struct stack *s,char ele)
{
    if (s->top < MAX-1)
    s->data[++s->top]=ele;
    else
    printf("\nSTACK OVERFLOW");
}

char pop(struct stack *s)
{
    if(!empty(s))
    return s->data[s->top--];
    else
    return (char)-1;
}

int precedence(char a)
{
    switch(a)
    {
        case '+':
        return 1;
        case '-':
        return 1;
        case '*':
        return 2;
        case '/':
        return 2;
        case '^':
        return 3;
        case '$':
        return 3;
```

```
    }
    return 0;
}

char *convert(char *expr)
{
    char *pexpr;
    char c,sc='\0';
    int i,j,n;
    struct stack st;
    st.top=-1;
    for(n=0;expr[n]!='\0';n++)
    pexpr=(char *)malloc(n+1);

    for(i=0,j=0;expr[i]!='\0';i++)
    {
        c=expr[i];
        switch(c)
        {
            case '+':
            case '-':
            case '*':
            case '/':
            case '^':
            case '$':
            while(!empty(&st)&&
            precedence(st.data[st.top])>=precedence(c))
            {
                sc=pop(&st);
                pexpr[j++]=sc;
            }
            push(&st,c);
            break;

            case '(':
            push(&st,c);
            break;
            case ')':
            while((sc=pop(&st))!='(')
            pexpr[j++]=sc;
            break;
            default:
            pexpr[j++]=c;
        }
    }
    while(!empty(&st))
    pexpr[j++]=pop(&st);
    pexpr[j]='\0';
    return pexpr;
}

int main()
{
    char *infix , *postfix;
    infix=(char*)malloc(1);
    printf("\nEnter an Infix Expression For converting it into a
    Postfix Expression: " );
    scanf("%s",infix);
    postfix=convert(infix);
```

```
116         printf("\nThe Postfix convesion is : %s\n",postfix);
117     }
118
```