```
1
 2
      Titttle:- CONVERSION OF INFIX TO POSTFIX.
 3
      Name: - TAUSEEF MUSHTAQUE ALI SHAIKH
      Roll no.:- 18C063
 4
 5
      Class:- S.Y. [CO]
Date:- 29-07-2019
 6
 7
      DISCRIPTION: - This Program is made to convert a Infix String
                                                                                ₽
      given by the user into Equivalent Postfix String.
8
 9
     #include<stdio.h>
10
     #include<stdbool.h>
     #include<unistd.h>
11
12
     #define MAX 50
13
14
15
     struct stack
16
17
18
     char data[MAX];
19
     int top;
20
     };
21
     bool empty(struct stack *s)
22
23
24
          return (s->top==-1)?true:false;
25
     }
26
27
     void push(struct stack *s,char ele)
28
29
         if (s->top < MAX-1)
30
         s->data[++s->top]=ele;
31
         else
32
         printf("\nSTACK OVERFLOW");
33
     }
34
35
     char pop(struct stack *s)
36
37
         if(!empty(s))
38
         return s->data[s->top--];
39
         else
40
          return (char)-1;
     }
41
42
43
     int precedence(char a)
44
45
         switch(a)
46
              case '+':
47
48
              return 1;
49
              case '-':
50
              return 1;
51
              case '*':
52
              return 2;
53
              case '/':
              return 2;
54
55
              case '^':
56
              return 3;
57
              case '$':
58
              return 3;
```

```
59
   60
             return 0;
        }
   61
   62
        char *convert(char *expr)
   63
   64
   65
             char *pexpr;
             char c,sc='\0';
   66
             int i,j,n;
   67
   68
             struct stack st;
   69
             st.top=-1;
   70
             for(n=0;expr[n]!='\0';n++)
   71
             pexpr=(char *)malloc(n+1);
   72
   73
             for(i=0, j=0; expr[i]!='\0'; i++)
   74
   75
                 c=expr[i];
   76
                 switch(c)
   77
                 {
                      case '+':
   78
                      case '-':
   79
                      case '*':
   80
                      case '/':
   81
                      case '^':
   82
                      case '$':
   83
                      while(!empty(&st)&&
   84
                                                                                     ₽
                      precedence(st.data[st.top])>=precedence(c))
   85
   86
                          sc=pop(&st);
   87
                          pexpr[j++]=sc;
   88
   89
                      push(&st,c);
   90
                      break;
   91
                      case '(':
   92
   93
                      push(&st,c);
   94
                      break;
   95
                      case
   96
                      while((sc=pop(&st))!='(')
   97
                      pexpr[j++]=sc;
                      break;
   98
   99
                      default:
  100
                      pexpr[j++]=c;
                 }
  101
  102
             }
             while(!empty(&st))
  103
             pexpr[j++]=pop(&st);
  104
  105
             pexpr[j]='\0';
  106
             return pexpr;
        }
  107
  108
        int main()
  109
  110
  111
             char *infix , *postfix;
  112
             infix=(char*)malloc(1);
             printf("\nEnter an Infix Expression For converting it into a
Postfix Expression: " );
  113
                                                                                     ₽
             scanf("%s",infix);
  114
  115
             postfix=convert(infix);
- 2 -
```

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