

Experiment No. 3

Name: Siddhesh Vinay Rane

Class: SY-IT

Roll no: 47

Program :

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>
#define SIZE 100
char stack[SIZE];
int top=-1;
void push(char item)
{ if(top >= SIZE-1)
  { printf("\nStack Overflow");
  }
  else
  { top=top+1;
    stack[top]=item;
  }
}
char pop()
{ char item;
  if(top <0)
  { printf("Stack Underflow: Invalid Infix Expression");
    getchar();
    exit(1);
  }
  else
  { item = stack[top];
    top = top-1;
    return(item);
  }
}
int operator(char sign)
{ if(sign=='^' || sign=='*' || sign=='/' || sign=='+' || sign=='-')
  { return 1;
  }
  else
  { return 0;
  }
}
int precedence(char sign)
{ if(sign=='^')
  { return(3);
  }
  else if(sign=='*' || sign=='/')
  { return(2);
  }
  else if(sign=='+' || sign=='-')
  { return(1);
  }
  else
  { return(0);
  }
}
```

```

    }
}
void inftopost(char infix_exp[],char postfix_exp[])
{ int i,j;
  char item,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(operator(item)==1)
    { x=pop();
      while(operator(x)==1&&precedence(x)>=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("\nInvalid infix Expression.\n");
      getchar();
      exit(1);
    }
    i++;
    item=infix_exp[i];
  }
  if(top>0)
  { printf("\nInvalid infix Expression.\n");
    getchar();
    exit(1);
  }
  postfix_exp[j]='\0';
}
int main()
{ char infix[SIZE],postfix[SIZE];
  printf("\n Enter Infix expression: ");
  fgets(infix,100,stdin);
  inftopost(infix,postfix);
  printf("Postfix Expression:");

```

Output:

