OUTPUT:

```
Enter valid parenthesized infix expression:2+4*(3-4/5)^9
Postfix Expression: 24345/-9^*+
PS C:\Users\student\Documents\1BF24CS195(DS Lab)> [
```

OBSERVATION:

```
Infix to Postfix
Q was to convert a given valid paramenessed in arthmetic expression
    to tastia capración the expression consists of single chareter
    operant and the binary operantors
        + (Mus), - (minus), + (multiply) and I (divide).
-> Step 1. We will define a stack
    Step 2 Define functions like pop (to dutite topeliment),
             bush (to add an element to stack) and kuck (to know
             first (top dumont of stack)
    Step 3: A function to know precidence of operators in defined
    step4 for same procedure operators, we define a function
              to know associativity of operators
     step 5: Then function to convert infin to postifix in write-
 Code (Bardo)
> void (pop) 1
        if (top == N-1) !
            bridf ("stack owylow");
         class ?
            topat;
            incoming eliment in addition stack
     1 (dod) bion
        il (fop==-1) 1
         elsi
             delete me top element
```

```
void (bed)
   to sum upeliment of stock
int preadence (chor ob) !
      switch (op)
          casi - relum 'J';
          cos * -
                                          ( 2) Hours
           cost 1' veturn 2;
           cox 'n' return3;
           casi ( setum 0;
           default return -1;
  int associativity (char ob) {
      if (ob== ~) {
            return 1;
       returno;
 int conversion (chor infra [3, char post fix [3)}
                                           CHICARTE SAND
     inti, k=0;
      for ( before 10) infra [1]=10; [++){
        c=infiz[i];
        if (isal num(c))}
              push (c)
          Elsi if ( == "1") {
                                        arthurth was
             bush (c);
           classif ((==')') [
                while (buk () ) = (') !
                   posty ( Had - Febl);
```

```
bob();
        class {
        while I topt - 198 presduce (beepl) presduce (0)) 11,
                              breadma (peck () > bread () 71
                               associativity (c)==0)))[
           bush (1)
     while (top1 >-1) |
            bostfin(k++)= bob();
     Code
# include (statio.h)
# include (ctype h)
# include (string h)
# define nax 100
Char stack [HAX]
int top= -1;
Char puch (chor c) {
     if (top== MAX-1) }
          brindf ("stock auglaul");
     clas f
          Hobiti)
         Stock [tob]=c;
```

```
Char peplas
    if (tob==-1){
        brith ( Stack underflow in );
        return -1,
                                      1990 ------
    return stock [tob--];
                                       17 37 31 11 1415
char puk () 1
    if (top == -1) H
3
int preadence (char op) {
    Switch (op) 1
     case the deposition of the
             break; a seed in the location
      (casi'+'
                  Charlet next separat
      (asi / :
            veturn 2;
             break,
      casi'n'
            veturn 3,
            break;
      con 10'
            return 0;
            break;
     return -1;
                             There apply very translated and
 int associativity (char ob) [
                       and the factor was talm or of Thomas
     If (ob== 'n')
          return!
     returno:
```

```
int conversion (char infix (), that postin ()) {
       int (, R=0,
       Char C;
       for (C=0, in fix[C])= 10'; (++) 1
           c= inhix [1];
            if (isalnum (c))
                 Rostha (KH)=0,
            ds if ( (== '('))
                  push (c);
              educif (c==')') {
                  while (peek () = '(') {
                      postfix [k++] = pop();
                   POP();
              elsif
                 while (top!=1 33 ((precidence (peute))) > brucidence (c))
                        11. (predena (kuk()) == preudena (c) 24
                              ossociativity (c) == 0 2) f.
                          postfix [m+)= popl);
                                                  S. TENSTON
        while (top)=-1)
               bostfix (RH) - pop();
 int maintal
     unur infix[HOL], char bookin [HOL];
     print) ("Giver valid parenthusised infin expression.");
     scanf (" "10", (m/2);
     conversion (infiniposthis);
      brinds (" Post fix Extrapolon: 7.5 \n', Fostfix);
```

Output -> 6nter Valid infin 6nteression: 2-4*(3^6+4)/8

Postfina 6nteression: 2436^4+*8/
Sour