06/10

WAP. to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

```
#include<stdio.h>
#include<ctype.h>
#define size 50
char stack[size];
int top = -1;
Void push(char element)
   { stack[++top] = element;
   }
Char pop() {
   return stack[top--];
}
int
pr(char symbol){
   if(symbol == '^')
     { return 3;
     }
   else if(symbol = '*' || symbol = '/')
     { return 2;
     }
   else if(symbol = '+' || symbol = '-')
     { return 1;
     }
   else
      return 0;
```

```
int main(){
 Char infix[50], Postfix[50], ch, element; Symbol
 int i=0, k=0;
 Printf("Enter the expression");
 get[s] ("%s", infix); push('#')
 while (ch= infix[i++]) != '\0'){
   if (ch= '(') {
       Push(ch);
   }
   else if (isalnum(unsigned char) ch)) {
       Postfix[k++] = ch;
   }
   else if (ch= ')') {
       while(stack[top]!= '(')
           Postfix[k++] = pop();
       element = pop();
       (void) element;
   }
  else {
   while (stack[top] != '(' && pr(stack[top])
          >= pr(ch))
       Postfix[k++] = pop();
     Push(ch);
  }
 while (stack[top] != '#')
     Postfix[k++] = pop();
   Post fix[k] = '\0';
   printf("\nPost fix = %s\n", Postfix);
  return 0;
}
```

output
Enter the In
Postfix = A

9/10

6/10

element

(stack[top])

POSTFIX);

output

Enter the Infix expression: (A+(B*C-CD/E^F)*G)*H)

Postfix = ABC*DEF^/G*-H*+