1. 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)

Program:

#include <stdio.h> #include <ctype.h> #define SIZE 50 char stack[SIZE]; int top = -1;

void push(char elem)

{

stack[++top] = elem;

}

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;

}

}

void main()

{

char postfix[50], infix[50], ch, elem; int i = 0, k = 0;

printf("Enter the infix expression: "); scanf("%s", infix);

push('#');

while ((ch = infix[i++]) != '\0')

{

if (ch == '(')

{

push(ch);

}

else if (isalnum(ch))

{

postfix[k++] = ch;

}

else if (ch == ')')

{

while (stack[top] != '(')

{

postfix[k++] = pop();

}

pop();

}

else

{

while (pr(stack[top]) >= pr(ch))

{

postfix[k++] = pop();

}

push(ch);

}

}

while (stack[top] != '#')

{

postfix[k++] = pop();

}

postfix[k] = '\0';

printf("Postfix expression = %s\n", postfix);

}

