**Infix to prefix conversion:**

#include<stdio.h>

#include<string.h>

#include<limits.h>

#include<stdlib.h>

#define MAX 100

int top = -1;

char stack[MAX];

int isFull ()

{

return top == MAX - 1;

}

int isEmpty ()

{

return top == -1;

}

void push (char item)

{

if (isFull ())

return;

top++;

stack[top] = item;

}

int pop ()

{

if (isEmpty ())

return INT\_MIN;

return stack[top--];

}

int peek ()

{

if (isEmpty ())

return INT\_MIN;

return stack[top];

}

int checkIfOperand (char ch)

{

return (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z');

}

int precedence (char ch)

{

switch (ch)

{

case '+':

case '-':

return 1;

case '\*':

case '/':

return 2;

case '^':

return 3;

}

return -1;

}

int getPostfix (char \*expression)

{

int i, j;

for (i = 0, j = -1; expression[i]; ++i)

{

if (checkIfOperand (expression[i]))

expression[++j] = expression[i];

else if (expression[i] == '(')

push (expression[i]);

else if (expression[i] == ')')

{

while (!isEmpty (stack) && peek (stack) != '(')

expression[++j] = pop (stack);

if (!isEmpty (stack) && peek (stack) != '(')

return -1;

else

pop (stack);

}

else

{

while (!isEmpty (stack)

&& precedence (expression[i]) <= precedence (peek (stack)))

expression[++j] = pop (stack);

push (expression[i]);

}

}

while (!isEmpty (stack))

{

expression[++j] = pop (stack);

expression[++j] = '\0';

}

}

void reverse (char \*exp)

{

int size = strlen (exp);

int j = size, i = 0;

char temp[size];

temp[j--] = '\0';

while (exp[i] != '\0')

{

temp[j] = exp[i];

j--;

i++;

}

strcpy (exp, temp);

}

void brackets (char \*exp)

{

int i = 0;

while (exp[i] != '\0')

{

if (exp[i] == '(')

exp[i] = ')';

else if (exp[i] == ')')

exp[i] = '(';

i++;

}

}

void InfixtoPrefix (char \*exp)

{

int size = strlen (exp);

reverse (exp);

brackets (exp);

getPostfix (exp);

reverse (exp);

}

int main ()

{

printf ("The infix is: ");

char expression[] = "((a/b)+c)-(d+(e\*f))";

printf ("%s\n", expression);

InfixtoPrefix (expression);

printf ("The prefix is: ");

printf ("%s\n", expression);

return 0;

}

**Infix to postfix conversion:**

#include<stdio.h>

#include<ctype.h>

char stack[100];

int top = -1;

void push(char x)

{

stack[++top] = x;

}

char pop()

{

if(top == -1)

return -1;

else

return stack[top--];

}

int priority(char x)

{

if(x == '(')

return 0;

if(x == '+' || x == '-')

return 1;

if(x == '\*' || x == '/')

return 2;

return 0;

}

int main()

{

char exp[100];

char \*e, x;

printf("Enter the expression : ");

scanf("%s",exp);

printf("\n");

e = exp;

while(\*e != '\0')

{

if(isalnum(\*e))

printf("%c ",\*e);

else if(\*e == '(')

push(\*e);

else if(\*e == ')')

{

while((x = pop()) != '(')

printf("%c ", x);

}

else

{

while(priority(stack[top]) >= priority(\*e))

printf("%c ",pop());

push(\*e);

}

e++;

}

while(top != -1)

{

printf("%c ",pop());

}return 0;

}

Answer for the snippets:

1.Error

2.44

3.80

4.E, 69

5.1