#include<stdio.h>

#include <string.h>

#include <ctype.h>

char operand\_stack[50][80], operator\_stack[50];

int top\_operator = -1, top\_operand = -1;

int push\_operator(char operator)

{

operator\_stack[++top\_operator] = operator;

}

int push\_operand(char \*opnd)

{

strcpy(operand\_stack[++top\_operand], opnd);

}

char pop\_operator()

{

return(operator\_stack[top\_operator--]);

}

char \*pop\_operand()

{

return(operand\_stack[top\_operand--]);

}

int empty(int t)

{

if(t == 0)

{

return 1;

}

else

{

return 0;

}

}

int main()

{

char prefix\_expression[50], ch, temporary\_string[50], operand\_a[50], operand\_b[50], operator[2];

int count = 0, k = 0, operand\_count = 0;

printf("\nEnter a Prefix Expression:\t");

scanf("%s", prefix\_expression);

while((ch = prefix\_expression[count++]) != '\0')

{

if(isalnum(ch))

{

temporary\_string[0] = ch;

temporary\_string[1]='\0';

push\_operand(temporary\_string);

operand\_count++;

if(operand\_count >= 2)

{

strcpy(operand\_b, pop\_operand());

strcpy(operand\_a, pop\_operand());

strcpy(temporary\_string, "(");

strcat(temporary\_string, operand\_a);

ch = pop\_operator();

operator[0] = ch;

operator[1] = '\0';

strcat(temporary\_string, operator);

strcat(temporary\_string, operand\_b);

strcat(temporary\_string, ")");

push\_operand(temporary\_string);

operand\_count = operand\_count - 1;

}

}

else

{

push\_operator(ch);

if(operand\_count == 1)

{

operand\_count = 0;

}

}

}

if(!empty(top\_operand))

{

strcpy(operand\_b, pop\_operand());

strcpy(operand\_a, pop\_operand());

strcpy(temporary\_string, "(");

strcat(temporary\_string, operand\_a);

ch = pop\_operator();

operator[0] = ch;

operator[1] = '\0';

strcat(temporary\_string, operator);

strcat(temporary\_string, operand\_b);

strcat(temporary\_string, ")");

push\_operand(temporary\_string);

}

printf("\nInfix Expression:\t %s\n", operand\_stack[top\_operand]);

return 0;

}