#include <stdio.h>

#include <malloc.h>

struct node

{

char \*base;

char \*top;

};

void push(struct node \*L,char x)

{

\*L->top=x;

L->top++;

}

char pop(struct node \*L)

{

char x;

if(L->top==L->base) x='0';

else

{

L->top--;

x=\*L->top;

}

return x;

}

int main()

{

char s[320],c;

int i=0;

struct node \*L;

L=(struct node \*)malloc(sizeof(struct node));

L->base=(char \*)malloc(320\*sizeof(char));

L->top=L->base;

gets(s);

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

{

if(s[i]>='a'&&s[i]<='z'||s[i]>='A'&&s[i]<='Z')

printf("%c",s[i]);

else if(L->base==L->top||s[i]=='(')

push(L,s[i]);

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

{

c=pop(L);

while(c!='(')

{

printf("%c",c);

c=pop(L);

}

}

else

{

if(s[i]=='+'||s[i]=='-')

{

c=pop(L);

while(c!='0'&&c!='(')

{

printf("%c",c);

c=pop(L);

}

if(c=='(') push(L,c);

push(L,s[i]);

}

else

{

c=pop(L);

while((c=='/'||c=='\*')&&c!='0'&&c!='(')

{

printf("%c",c);

c=pop(L);

}

if(c=='+'||c=='-'||c=='(') push(L,c);

push(L,s[i]);

}

}

i++;

}

c=pop(L);

while(c!='0')

{

printf("%c",c);

c=pop(L);

}

return 0;

}

**问题 M: 中缀表达式转后缀表达式**

时间限制: 1 Sec 内存限制: 128 MB  
提交: 2077 解决: 625  
[[提交](http://acm.hnust.cn/JudgeOnline/submitpage.php?cid=1569&pid=12&langmask=0)][[状态](http://acm.hnust.cn/JudgeOnline/problemstatus.php?id=1823)][[讨论版](http://acm.hnust.cn/JudgeOnline/bbs.php?pid=1823&cid=1569)]

**题目描述**

输入一个中缀表达式，编程输出其后缀表达式，要求输出的后缀表达式的运算次序与输入的中缀表达式的运算次序相一致。为简单起见，假设输入的中缀表达式由＋（加）、－（减）、×（乘）、／（除）四个运算符号以及左右圆括号和英文字母组成，其中算术运算符遵守先乘除后加减的运算规则。假设输入的中缀表达式长度不超过300个字符，且都是正确的，即没有语法错误，并且凡出现括号其内部一定有表达式，即内部至少有一个运算符号。  
  
中缀表达式转后缀表达式的方法：  
1.遇到操作数：直接输出（添加到后缀表达式中）  
2.栈为空时，遇到运算符，直接入栈  
3.遇到左括号：将其入栈  
4.遇到右括号：执行出栈操作，并将出栈的元素输出，直到弹出栈的是左括号，括号不输出。  
5.遇到其他运算符：加减乘除：弹出所有优先级大于或者等于该运算符的栈顶元素，然后将该运算符入栈  
6.最终将栈中的元素依次出栈，输出。

**输入**

只有一行，为中缀表达式

**输出**

只有一行，为转换后的后缀表达式

**样例输入**

X+A\*(Y-B)-Z/F

**样例输出**

XAYB-\*+ZF/-