当前作业

» 22级第六次作业（查找与排序）

» 22级第五次作业（树）

» 2022级（信息大类）数据结构综合作业（正确性和性能）

» 2022级（信息大类）数据结构综合作业（可扩展性）

历史作业

» 22级第四次作业（栈和队）

» 22级第三次作业（线性表）

» 22级第二次作业

» 22级第一次作业

» 21级第七次作业（图）

» 21级第六次作业（查找与排序）

» 21级第五次作业（树）

» 2021级（信息大类）数据结构综合作业（正确性和性能）

» 2021级（信息大类）数据结构综合作业（可扩展性）

» 21级第四次作业（栈和队）

» 21级第三次作业

21级第四次作业（栈和队）

作业时间： **2022-04-14 18:00:00** 至 **2022-07-01 02:00:00**

本次作业主要考查对栈和队知识的掌握情况，请用相关知识完成本次作业。

选择题

1. 首次提交时间:2022-04-14 18:55:07 最后一次提交时间:2022-04-14 18:55:07

栈和队都是C

A. 顺序存储的线性结构

B.链式存储的非线性结构

C. 限制存取点的线性结构

D.限制存取点的非线性结构

2. 首次提交时间:2022-04-14 18:58:32 最后一次提交时间:2022-04-14 18:58:44

递归过程或函数调用时，处理参数及返回地址，要用一种称为C 的数据结构。

A. 队列

B. 多维数组

C. 栈

D. 线性表

3. 首次提交时间:2022-04-14 19:08:27 最后一次提交时间:2022-04-14 19:08:27

设栈S和队列Q的初始状态为空，元素e1，e2，e3，e4,e5和e6依次通过栈S，一个元素出栈后即进队列Q，若6个元素出队的序列是e2，e4，e3,e6,e5,e1则栈S的容量至少应该是C

A.6

B.4

C.3

D.2

4. 首次提交时间:2022-04-14 19:09:04 最后一次提交时间:2022-04-14 19:09:04

设一个栈的输入序列是 1，2，3，4，5,则下列序列中，是栈的合法输出序列的是
D

A. 5 1 2 3 4

B. 4 5 1 3 2

C. 4 3 1 2 5

D. 3 2 1 5 4

5. 首次提交时间:2022-04-14 19:09:26 最后一次提交时间:2022-04-14 19:09:26

一个栈的进栈序列是a，b，c，d，e，则栈的不可能的输出序列是 C 。

A．edcba B．decba C．dceab D．abcde

6. 首次提交时间:2022-04-14 19:10:00 最后一次提交时间:2022-04-14 19:10:00

中缀表达式A-(B+C/D)×E的后缀形式是 B 。

A . ABC+D/ ×E- B. ABCD/+E×- C. AB-C+D/E× D. ABC-+D/E×

7. 首次提交时间:2022-04-14 19:14:08 最后一次提交时间:2022-04-14 19:14:08

在非空双向循环链表中由q所指的那个链结点前面插入一个由p所指的链结点的动作所对应的语句依次为：p—>rlink=q; p—>llink=q—>llink; q—>llink=p; D 。(空白处为一条赋值语句)

A. q—>rlink= p;

B. q—>llink—>rlink=p;

C. p—>rlink—>rlink= p;

D. p—>llink—>rlink=p;

8. 首次提交时间:2022-04-14 19:12:56 最后一次提交时间:2022-04-14 19:12:56

【单选题】
若栈和队都采用顺序存储结构，则下述说法正确的是C 。

A. 任何情况下都可以进行出栈操作。
B. 任何情况下都可以进行进栈操作。
C. 队不为空时可以进行出队操作。
D. 任何情况下都可以进行入队操作。

9. 首次提交时间:2022-04-14 19:14:21 最后一次提交时间:2022-04-14 19:14:21

为解决计算机主机与打印机之间速度不匹配问题，通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是B 。

A．栈 B．队列 C．树 D．图

10. 首次提交时间:2022-04-14 19:16:55 最后一次提交时间:2022-04-14 19:16:55

己知循环队列存储在一维数组A[0-n-1]中，且队列非空时front和rear分别指向队头元素和队尾元索。若初始时队列为空，且要求第1个进入队列的元素存储在A[0]处，则初始时front和rear的值分别是
B 。

A.0,0 B.0,n-1 C.n-1,0 D.n-1,n-1

11. 首次提交时间:2022-04-14 19:17:04 最后一次提交时间:2022-04-14 19:17:04

允许对队列进行的操作有D 。

A．对队列中的元素排序 B．取出最近进队的元素 C．在队头元素之前插入元素 D．删除队头元素

12. 首次提交时间:2022-04-14 19:17:36 最后一次提交时间:2022-04-14 19:17:36

【单选题】
设有一顺序栈S，元素a,b,c,d,e,f,g,h依次进栈，如果8个元素出栈的顺序是d,f,e,c,h,g,b,a，则栈的容量至少应该是C

A、3　　B、4　　C、5　　D、6

13. 首次提交时间:2022-04-14 19:22:42 最后一次提交时间:2022-04-14 19:22:42

若用一个大小为6的数组来实现循环队列，且当前rear和front的值分别为0和3，当从队列中删除一个元素，再加入两个元素后，rear和front的值分别为多少？B

A. 1和5

B. 2和4

C. 4和2

D. 5和1

填空题

1. 首次提交时间:2022-04-14 19:23:21 最后一次提交时间:2022-04-14 19:24:09 已提交

下列程序判断字符串s 是否对称，对称则返回1，否则返回0；如 f("abba")返回1，f("abab")返回0；

```
int f(___char *s___)
{
    int   i=0,j=0;
    while (s[j])___++j___;
    for(j--; i<j  && s[i]==s[j]; i++,j--);
    return(___i>=j___) ;
}
```

2. 首次提交时间:2022-04-14 19:24:32 最后一次提交时间:2022-04-14 19:24:58 已提交

用S表示入栈操作，X表示出栈操作，若元素入栈的顺序为1234，为了得到1342出栈顺序，相应的S和X的操作串为___SXSSXSXX___

3. 首次提交时间:2022-04-14 19:25:24 最后一次提交时间:2022-04-14 19:25:31 已提交

若已知一个栈的入栈序列是1,2,3...，30，其输出序列是p1,p2,p3,...pn, 若p1=30, 则p10为___21___。

4. 首次提交时间:2022-04-14 19:28:20 最后一次提交时间:2022-04-14 19:28:28 已提交

若某栈初始为空，PUSH与POP分别表示对栈进行一次进栈与出栈操作，那么对于进栈序列a,b,c,d,e,经过PUSH,PUSH,POP,PUSH,POP,PUSH,PUSH以后，得到的出栈序列是___b,c___。（答案用","隔开，如：a,b,c）。

5. 首次提交时间:2022-04-14 19:25:53 最后一次提交时间:2022-04-14 19:28:04 已提交

中缀表达式3+x*(2.4/5-6)所对应的后缀表达式为___3 x 2.4 5 / 6 - * +___

6. 首次提交时间:2022-04-14 19:27:21　最后一次提交时间:2022-04-14 19:27:33　已提交

栈R,从顶到底:{2,4,6,8,10},逐个取出放入队列Q中，再从Q中逐个取出放入R中，问现在栈R中从顶到底的顺序为 ___{10,8,6,4,2}___。
输出格式：{1,2,3,4,5}

7. 首次提交时间:2022-04-14 19:25:41　最后一次提交时间:2022-04-14 19:25:41　已提交

描述某循环队列的数组为QUEUE[0..M-1]，当循环队列满时，队列中有 ___M___个元素。

编程题

#	题目	分值	批阅信息
1.	栈操作（栈-基本题）	20.00	下载源文件

得分20.00　最后一次提交时间:2022-04-14 18:55:39

共有测试数据:5
平均占用内存:1.397K　平均CPU时间:0.00503S　平均墙钟时间:0.00502S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	完全正确
测试数据5	完全正确

详细

【问题描述】

假设给定的整数栈初始状态为空，栈的最大容量为100。从标准输入中输入一组栈操作，按操作顺序输出出栈元素序列。栈操作：1表示入栈操作，后跟一个整数（不为1、0和-1）为入栈元素；0表示出栈操作；-1表示操作结束。

【输入形式】

从标准输入读取一组栈操作，入栈的整数和表示栈操作的整数之间都以一个空格分隔。

【输出形式】

在一行上按照操作的顺序输出出栈元素序列，以一个空格分隔各元素，最后一个元素后也要有一个空格。如果栈状态为空时进行出栈操作，或栈满时进行入栈操作，则输出字符串“error”，并且字符串后也要有一空格。所有操作都执行完后，栈也有可能不为空。

【样例输入】

```
1 3 1 5 1 7 0 0 1 8 0 1 12 1 13 0 0 0 1 90 1 89 0 -1
```

【样例输出】

```
7 5 8 13 12 3 error 89
```

【样例说明】

入栈元素依次为3、5、7，然后有两次出栈动作，所以先输出7和5，这时栈中只有元素3；之后元素8入栈，又出栈，输出8；随后元素12和13入栈，再进行4次出栈操作，输出13、12和3，这时栈为空，再进行出栈操作会输出error；最后90和89入栈，进行一次出栈操作，输出89，栈中剩余1个元素。

【评分标准】

该题要求按照操作的顺序输出出栈元素序列，提交程序名为stack.c。

#	题目	分值	批阅信息
2.	C程序括号匹配检查	20.00	下载源文件

【问题描述】

编写一程序检查C源程序文件中{}、()等括号是否匹配,并输出第一个检测到的不匹配的括号及所对应括号所在的行号（程序中同一类括号只有一个不匹配）。

注意：

1. 除了括号可能不匹配外，输入的C源程序无其它语法错误。

2. 字符常量、字符串常量及注释中括号不应被处理，注释包括单行注释//和多行/* */注释

3. 字符常量和字符串常量中不包含转义字符\'和\"；

4. 程序中出现有意义括号的个数不超过200个；

不匹配判断规则：

1. 当检测的程序括号为'{'时，若其前序尚未匹配的括号为'('时，输出该'('左括号及所在行号；

2. 当遇到一个不匹配的右括号')'或'}'时，输出该右括号及所在行号；

3. 当程序处理完毕时，还存在不匹配的左括号时，输出该左括号及所在行号。

【输入形式】

打开当前目录下文件example.c，查询其括号是否匹配。该文件中每行字符数不超过200。

【输出形式】

若存在括号不匹配时，应输出首先能判断出现不匹配的括号及其所在的行号。当出现括号不匹配时，按下面要求输出相关信息：

without maching <x> at line <n>

其中<x>为'{', '}', '(', ')'等符号，<n>为该符号所在的行号。

若整个程序括号匹配，则按下面所示顺序输出括号匹配情况，中间没有空格。

(){(()){}}

【样例输入1】

若当前目录下输入文件example.c中内容如下：

#	题目	分值	批阅信息

#include<stdio.h>

int main(){

printf("{ hello world }\n"); // }

)

【样例输出1】

without maching ')' at line 4

【样例输入2】

若当前目录下输入文件example.c中内容如下：

#include<stdio.h>

int main(){

printf("{ hello world }d\n"); /* }*/

【样例输出2】

without maching '{' at line 2

【样例输入3】

若当前目录下输入文件example.c中内容如下：

#include<stdio.h>

int main(){

printf("{ hello world }d\n"); /* }*/

}

【样例输出3】

(){()}

【样例说明】

样例1：在注释部分和字符串中的括号不考虑，在将程序处理之后得到的括号序列是（）{（）），遇到右括号时与最近的左括号匹配，发现最后一个小括号和大括号不匹配。

样例2：处理之后的括号序列是（）{（），在最后缺少了右大括号，那么应该输出与之相对应的左括号不匹配。

【评分标准】

通过所有测试点得满分。

得分20.00　最后一次提交时间:2022-04-16 15:13:24

共有测试数据:5

平均占用内存:1.398K　平均CPU时间:0.00536S　平均墙钟时间:0.00534S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	完全正确
测试数据5	完全正确

详细

#	题目	分值	批阅信息
3.	计算器（表达式计算-后缀表达式实现，结果为浮点）	20.00	下载源文件

【问题描述】

从标准输入中读入一个算术运算表达式，如：24 / (1 + 5/3 + 36 / 6 / 2 - 2) * (12 / 2 / 2)= ，计算表达式结果，并输出。

要求：

1、表达式运算符只有+、-、*、/，表达式末尾的=字符表示表达式输入结束，表达式中可能会出现空格；
2、表达式中会出现圆括号，括号可能嵌套，不会出现错误的表达式；

3、表达式中出现的操作数都是十进制整数常量；但要求运算结果为浮点型，例如：5/2结果应为2.5。

4、要求采用逆波兰表达式来实现表达式计算。

【输入形式】

从键盘输入一个以=结尾的算术运算表达式。操作符和操作数之间可以有空格分隔。

【输出形式】

在屏幕上输出计算结果，小数点后保留两位有效数字。

【样例输入】

24 / (1 + 5/3 + 36 / 6 / 2 - 2) * (12 / 2 / 2) =

【样例输出】

19.64

【样例说明】

按照运算符及括号优先级依次计算表达式的值。

【评分标准】

该题要求采用逆波兰表达式实现表达式运算，提交程序名为cal.c。

得分20.00　最后一次提交时间:2022-04-15 15:27:03

成功编译,但有警告信息.

```
cal.c: In function 'opt_cmp':
cal.c:32:5: warning: array subscript has type 'char' [-Wchar-subscripts]
else return h[a]-h[b];
^
cal.c:32:5: warning: array subscript has type 'char' [-Wchar-subscripts]
cal.c: In function 'main':
cal.c:102:5: warning: implicit declaration of function 'gets' [-Wimplicit-function-declaration]
gets(epr);
^
/tmp/ccoS6kfA.o: In function `main':
cal.c:(.text.startup+0x3e): warning: the `gets' function is dangerous and should not be used.
```

共有测试数据:5

平均占用内存:1.400K　平均CPU时间:0.00502S　平均墙钟时间:0.00500S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	完全正确
测试数据5	完全正确

详细

#	题目	分值	批阅信息
4.	文本编辑操作模拟（简）a	20.00	下载源文件

【问题描述】

编写一程序模拟文本编辑操作。首先从标准输入读取一行字符串（字符个数不超过512），该行字符串是已经过n（大于0，小于等于10）步编辑操作后的结果。然后从下一行读取n，以及已发生过的n步编辑操作，编辑操作分行输入，输入格式为：

op pos str

其中op为编辑操作命令编码（在此只有插入和删除操作，1表示插入或2表示删除操作）；pos表示插入或删除的位置；str表示已经插入或删除的字符串（中间没有空格）。各数据间以一个空格分隔。

然后在空一行后，再分行输入当前将要进行的编辑操作，包括如下四种操作（操作编码分别为：1表示插入，2表示删除操作，3表示撤销（即undo操作），-1表示结束）：

1 pos str

表示将在pos位置插入字符串str（中间没有空格），各数据间以一个空格分隔；

2 pos n

表示将从pos位置开始删除n个字符（各数据间以一个空格分隔），若要删除的字符个数多于已有字符个数（即在文本中从pos开始的字符个数小于n），则按实际字符数删除即可。（提示：为了能够撤销删除操作，应按"2 pos str"形式保存命令。）

3

表示撤销最近执行的插入或删除操作，可以进行多次撤销操作，注意：也可以撤销之前已经发生过的n步编辑操作中的操作。

-1

表示退出编辑操作，在屏幕上输出最终编辑后的文本。

要求：

1、上述所有输入的编辑操作中的字符串str都不包含空白字符（空格符、制表符或换行符）；

2、插入操作中的位置pos大于等于0，并且小于等于当前文本的字符个数；0位置表示文本第一个字符的位置；若pos为当前文本的字符个数，则表示在文本最后插入字符串；

3、删除操作中的位置pos大于等于0，并且小于当前文字的字符个数；

得分20.00　最后一次提交时间:2022-04-15 17:17:58

成功编译,但有警告信息.

edit.c: In function 'main':

edit.c:34:5: warning: implicit declaration of function 'gets' [-Wimplicit-function-declaration]

gets(text);

^

/tmp/ccMGZWv1.o: In function `main':

edit.c:(.text.startup+0x18): warning: the `gets' function is dangerous and should not be used.

共有测试数据:5

平均占用内存:1.400K　平均CPU时间:0.00518S　平均墙钟时间:0.00515S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	完全正确
测试数据5	完全正确

详细

#	题目	分值	批阅信息

4、若已无操作可撤销，则再进行撤销操作无效；

5、文本在编辑过程中，总字符个数不会超过512。

【输入形式】

先从键盘输入一行字符串，表示已经经过n步编辑操作后的文本串，然后在下一行输入一个正整数n，并分行输入n步插入或删除操作（表示按时间先后顺序已进行的操作），格式如上所述。随后空一行，再分行输入将要进行的编辑操作，格式如上所述。直到输入-1操作为止。

【输出形式】

在屏幕上输出最终编辑后的文本内容。

【样例输入】

A Stack is a container of objects that are inserted and removed according to the l ast-in first-out (LIFO) principle.???

4

1 20 ainer

2 0 ???

1 85 -

1 99 (LIFO)

3

2 110 10

1 110 Objects

2 98 1

2 0 1

2 108 10

3

3

3

-1

【样例输出】

A Stack is a container of objects that are inserted and removed according to the l ast-in first-out principle.Objects

【样例说明】

第一行输入的文本串是先后经过下面4次编辑操作后得到的：先在20位置插入了字符串ainer，然后删除了开始位置的字符串???，随后在85位置插入了一个字符-，最后在99位置插入了字符串(LIF O)。

#	题目	分值	批阅信息
	随后输入了撤销操作，即撤销先前最后进行的“1 99 (LIFO)”操作，也就是将99位置的6个字符删除； 2 110 10：将文本串最后的字符串???删除； 1 110 Objects：在文本串末尾插入字符串Objects； 随后执行了三次删除操作，又执行了三次撤销操作，最后输入的-1表示编辑操作结束，在屏幕上输出最终编辑后的文本串。 【评分标准】 该程序要求编程模拟编辑操作，提交程序文件名为edit.c。		

#	题目	分值	批阅信息
5.	银行排队模拟（生产者-消费者模拟）	20.00	下载源文件

【问题描述】

一个系统模仿另一个系统行为的技术称为模拟，如飞行模拟器。模拟可以用来进行方案论证、人员培训和改进服务。计算机技术常用于模拟系统中。

生产者-消费者（Server-Custom）是常见的应用模式，见于银行、食堂、打印机、医院、超等提供服务和使用服务的应用中。这类应用的主要问题是消费者如果等待（排队）时间过长，会引发用户抱怨，影响服务质量；如果提供服务者（服务窗口）过多，将提高运管商成本。（经济学中排队论）

假设某银行网点有五个服务窗口，分别为三个对私、一个对公和一个外币窗口。银行服务的原则是先来先服务。通常对私业务人很多，其它窗口人则较少，可临时改为对私服务。假设当对私窗口等待服务的客户（按实际服务窗口）平均排队人数超过（大于或等于）7人时，等待客户将可能有抱怨，影响服务质量，此时银行可临时将其它窗口中一个或两个改为对私服务，当客户少于7人时，将立即恢复原有业务。设计一个程序用来模拟银行服务。

说明：

1. 增加服务窗口将会增加成本或影响其它业务，因此，以成本增加或影响最小为原则来增加服务窗口，即如果增加一个窗口就能使得按窗口平均等待服务人数小于7人，则只增加一个窗口。一旦按窗口平均等待服务人数小于7人，就减少一个所增加的窗口。

2. 为了简化问题，假设新到客户是在每个服务周期开始时到达。

3. 当等待服务人数发生变化时（新客户到达或有客户已接受服务），则及时计算按实际服务窗口平均等待服务人数，并按相应策略调整服务窗口数（增加或减少额外的服务窗口，但对私窗口不能减少）。注意：只在获取新客户（不管到达新客户数是否为0）时或已有客户去接受服务时，才按策略调整服务窗口数。进一步讲，增加服务窗口只在有客户到达的周期内进行（也就是说增加窗口是基于客户的感受，银行对增加窗口是不情愿的，因为要增加成本，一旦不再有新客户来，银行是不会再增加服务窗口的）；一旦有客户去接受服务（即等待客户减少），

得分20.00　最后一次提交时间:2022-04-15 15:21:32

共有测试数据:5

平均占用内存:4.680K　平均CPU时间:0.00541S　平均墙钟时间:0.00539S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	完全正确
测试数据5	完全正确

详细

#	题目	分值	批阅信息

银行将根据策略及时减少服务窗口，因此，在每个周期内，有客户去接受服务后要马上判断是否减少服务窗口（因为能减少成本，银行是积极的）

本问题中假设对公和对外币服务窗口在改为对私服务时及服务期间没有相应因公或外币服务新客户到达（即正好空闲），同时要求以增加成本或影响最小为前提，来尽最大可能减少对私服务客户等待时间。

【输入形式】

首先输入一个整数表示时间周期数，然后再依次输入每个时间周期中因私业务的客户数。注：一个时间周期指的是银行处理一笔业务的平均处理时间，可以是一分钟、三分钟或其它。例如：

6

2 5 13 11 15 9

说明：表明在6个时间周期内，第1个周期来了2个（序号分别为1,2），第2个来了5人（序号分别为3,4,5,6,7），以此类推。

【输出形式】

每个客户等待服务的时间周期数。输出形式如下：

用户序号 :等待周期数

说明：客户序号与等待周期数之间用符号:分隔，冒号（:）两边各有一个空格，等待周期数后直接为回车。

【样例输入】

4

2 5 13 11

【样例输出】

1 : 0

2 : 0

3 : 0

4 : 0

5 : 0

6 : 1

7 : 1

8 : 0

#	题目	分值	批阅信息

9 : 1

10 : 1

11 : 1

12 : 1

13 : 2

14 : 2

15 : 2

16 : 3

17 : 3

18 : 3

19 : 4

20 : 4

21 : 3

22 : 4

23 : 4

24 : 4

25 : 5

26 : 5

27 : 5

28 : 6

29 : 6

30 : 6

31 : 7

【样例说明】

样例输入表明有四个时间周期，第一个周期来了2人（序号1-2）；第二个周期来了5人（序号3-7）；第三个周期来了13人（序号8-20）；第四个周期来了11人（序号21-31）。由于第一个时间周期内只来了2人，银行（有三个服务窗口）能及时提供服务，因此客户等待时间为0；第二个时间周期内来了5人，银行一个周期内一次只能服务3人，另有2个在下个周期内服务，因此等待时间为1，其它类推。

【评分标准】

通过所有测试点得满分。

#	题目	分值	批阅信息
6.	函数调用关系（选做，不计分）	0.00	下载源文件

【问题描述】

给定某能正常运行结束的用户函数调用栈信息（当一个函数被调用时将入栈，当调用返回时，将出栈）。编写程序，对函数调用栈信息进行分析，依据函数入栈和出栈信息，分析函数调用关系，即一个函数调用了哪些不同函数。并按运行时调用序输出调用关系。

说明：

1. 在一个函数中，同一函数有可能被调用多次，输出调用关系时只输出一次；若一个函数没有调用其它函数，则不输出调用关系；

2. 函数运行时调用序是指函数在调用栈中的出现序。

3. 程序中不存在递归调用。函数名符合C语言标识符的规定，函数名长度不超过20，每个函数最多调用不超过10个不同函数，程序中用户定义的函数个数不超过100。

算法提示：当一个函数入栈时，它就是当前栈顶函数调用的一个函数。

【输入形式】

假设用8表示函数入栈操作；用0表示当前函数出栈。当操作为8（入栈）时，输入形式为：

<操作><函数名>

当操作为0（出栈）时，输入形式为：

<操作>

所有入栈操作和出栈操作都是从标准输入分行输入，假设调用栈中函数个数最多不超过200。开始时，调用栈为空，当调用栈再次为空时，输入结束。

【输出形式】

按运行时调用先后顺序输出函数调用关系到标准输出，每行为一个函数的调用关系信息，包括：函数名及被调用函数，函数与被调用函数间用一个英文冒号“:”分隔，被调用函数间用一个英文逗号“,”分隔，最后一个函数名后跟一个回车。若一个函数没有调用其它函数，则不输出。

【样例输入】

```
8 main
8 input
0
8 mysqrt
0
```

得分0.00　最后一次提交时间:2022-04-17 09:06:56

共有测试数据:5

平均占用内存:1.401K　平均CPU时间:0.00422S　平均墙钟时间:0.00423S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	输出错误
测试数据4	输出错误
测试数据5	完全正确

详细

#	题目	分值	批阅信息

8 findA

0

8 findB

8 area

8 mysin

0

8 mycos

0

8 mysqrt

0

0

0

8 findC

8 area

8 mysin

0

0

8 mysqrt

8 max

0

0

0

8 output

0

0

【样例输出】

main:input,mysqrt,findA,findB,fin
dC,ouput

mysqrt:max

findB:area

area:mysin,mycos,mysqrt

findC:area,mysqrt

【样例说明】

按照运行时调用函数的先后顺序，依次输出了main、mysqrt、findB、area和findC的函数调用关系。其中main函数调用了6个函数，按照运行时调用序依次输出。注意：mysqrt函数先于findB等函数出现在栈中，虽然mysqrt调用max较晚，但要先输出其调用关系。

【评分标准】

该题要求对函数调用栈信息进行分析，提交程序名为stack.c

北京航空航天大学

若重置密码，请与当前的任课教师联系