The C Programming Language Exercise

关于作业：

各位小伙伴，协会的第二次培训已经结束了。相信你已经更了解了我们，也对 C 语言有了更深层次的了解。可能你会觉得有点难，但毕竟万事开头难，希望你们能坚持下去，好好打基础，勿在浮沙筑高台。
这是第二次培训的配套练习，大家能先尝试一下。**如对题目有任何疑问都可以在群里或者联系群里的学长们提出。**
请尽量在 **11 月 14 日**之前写完，写好后请以邮件附件的形式发送到 homework@vidar.club ，**源码编码规范必须采用 UTF-8 。附件须为压缩包，压缩包命名格式 "C2*你的名字*" ，如 "C2*张三*.zip" 。该压缩包须包含每道题的 C 代码文件，命名格式 "C2*题号名字*" ，如第 1 题 "C2_01*张三.c*" 、*第 2 题第 1 小题 "C2_02(01)*张三.c" 。**
注意：本周的题目有部分函数题，请你提交这个函数以及可执行的调用代码。
在代码文件中也可以写下你的疑问和想法。学长们会抽时间将每份代码看一遍并回复你的。
可能你会觉得完成所有的题目有点吃力，也没有关系，尽自己的能力去完成就好，如果感觉学习阶段有了瓶颈，也能写明自己的问题给我们。不过我们想说明一点，我们希望你能学会独立思考，如果不会做，你可以选择参考别人的代码，但不要直接抄。直接抄来的东西不是你的，只有经过了你自己思考的才有意义。

1、Liki的大数库

Liki学姐非常喜欢密码学，但是密码学中，经常会涉及到一些非常非常非常大的数（比long long long int还要大的多！）的**精确计算**，你能否帮助她实现一个大数运算的库呢？（在你的库里，你依然可以调用别的c语言常规库）

（1）

输入：大数 大数
（P.S. 其中位数保证小于100位）
输出：两个大数的和：num1+num2

例：

输入

```
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
2222222222
```

输出

```
22233333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333
33333333333
```

注意：以下的调用程序仅供参考，你也可以递交你自己的调用程序

```
#include ""//导入你自己的库及必要的库

int main(){
    bignumber num1;
    bignumber num2;

    num1 = Scanbignumber();
    num2 = Scanbignumber();

    Printbignumber(Sumbignumber(num1,num2));
    return 0;
}
```

（2）除了加法之外，是否还可以实现大数乘法呢?

例:

输入：大数 大数
（P.S. 其中位数保证小于100位)
输出：两个大数的积：num1*num2

输入

```
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111
22222222222222222222222222222222222222222222222222222222222222222222222222222222
22222222222
```

输出

```
24691358024691358024691358024691358024691358024691358024691358024691358024691358
02469135555308641975308641975308641975308641975308641975308641975308641975308641
975308641975308642
```

注意：以下的调用程序仅供参考，你也可以递交你自己的调用程序

```
#include ""//导入你自己的库及必要的库

int main(){
    bignumber num1;
    bignumber num2;

    num1 = Scanbignumber();
    num2 = Scanbignumber();

    Printbignumber(Mulbignumber(num1,num2));
    return 0;
}
```

2、Trotsky的井字棋

Trotsky最近迷上了下井字棋，井字棋的游戏规则很简单，一方是X，另一方是O，执O的先下，在3*3的格子中轮流下棋，率先三子连成一线（横、竖、斜）者获胜。
Trotsky在挑战一个超强AI，它绝不会有失误，会竭尽全力不让你获胜。
现在有一些电脑随机生成的“残局”，Trotsky现在执O，你能否写一个程序，帮Trotsky判断两步之内能否获胜。
当然，你还需要判断当前这个“残局”是否能够继续游戏，或是否是一种不可能情况。

例：

输入：

```
ox_
ox_
xo_
```

输出：

```
can't win
```

输入：

```
_ox
_ox
___
```

输出：

```
can win
```

输入：

```
_oo
_xo
o_x
```

输出：

```
wrong
```

3、指针找对象

有26个全局指针，用A ~ Z表示，初始时指向各自的对象（如 A指向a）
有26个对象，用a ~ z表示；
每个对象还有26个成员变量a ~ z（如 a的成员a 表示成a.a）,它们也是指针，可以指向其他对象。

有四种指令

- A = x， A指向对象x
- A = B， A指向B指向的对象
- A.f = B， A指向的对象的成员变量f指向B指向的对象
- A = B.f， A指向B指向的对象的成员变量f指向的对象

给出n条随机的指令，请问这些指令按顺序执行后，每个全局指针指向的对象。

输入：
n
后面跟着n条指令，用回车符分隔

输出：
A -> ...
B -> ...
...

例：

输入：

```
4
A = b
A = C
A.c = D
E = C.c
```

输出：

```
A -> c
B -> b
C -> c
D -> d
E -> d
F -> f
G -> g
H -> h
I -> i
J -> j
K -> k
L -> l
M -> m
N -> n
O -> o
P -> p
Q -> q
R -> r
S -> s
T -> t
U -> u
V -> v
W -> w
X -> x
Y -> y
Z -> z
```

4、Mezone的烦恼

Mezone是一个优等生，每次评选奖学金时，他都要计算一番自己的平均成绩，你能否帮他写一个函数，让他能够很方便地计算自己的平均成绩。
需要注意的是：每个学期的科目数量都不一样多，所以请你使用**变参函数**来解决这个问题。
（1）先用 n 表示课程的个数，后面是各科的成绩

```
#include <stdio.h>

double ave(int n,...);//n代表可变部分总共有几个变量

int main(){
    printf("First term : %.2lf", ave(5, 88, 99, 70, 95, 93));
    printf("Second term : %.2lf", ave(7, 100, 99, 92, 88, 68, 94, 92));

}

/* 你的代码将会镶嵌到这 */
```

（2）每次都要数一数课程的个数（上题中的 n ）再平均有点麻烦，你能不能省略掉这个环节呢？如果能，给出你的代码，如果不能，给出你的解释。

```
#include <stdio.h>

/* 你的代码将会镶嵌到这 */

int main(){
    printf("First term : %.2lf", ave(88, 99, 70, 95, 93));
    printf("Second term : %.2lf", ave(100, 99, 92, 88, 68, 94, 92));

}
```

5、[附加题]

常量不可修改，这是 C 语言中的常识，但并非绝对性的结论。想要探寻真理，自然需要向着真理所在的更深领域前行，这题需要的知识超出了 C 语言的范畴，但充满好奇心的你未必不能找到答案。加油，也许这题就会是开启你二进制安全之旅的大门！

题目要求：在省略处补全代码，修改字符串常量，只更改输出的字符串，但 addr 不变。

Hint：Windows 下与 Linux 下的答案并不一致，若你在其中一种环境下成功，也可以试试寻找另一个环境下的答案。

```c
#include <stdio.h>
#include ......
/* 你可以继续include */
char*str = "I'm the most evil E99p1ant!!!";

int main()
{
    printf("initial addr:\n%p\ninitial E99p1ant:\n%s\n\n", str, str);
    ......
    ......
    printf("now addr:\n%p\nnow E99p1ant:\n%s\n",str,str);
    return 0;
}
```

参考成功输出：

```
initial addr:
00007FF6A38A9BB8
initial E99p1ant:
I'm the most evil E99p1ant!!!

now addr:
00007FF6A38A9BB8
now E99p1ant:
I'm the most hentai E99p1ant!!!
```