

# C7-Statement

这是 2021级-航类-C语言程序设计 的第七次上机。

- 上机时间2022/05/11 19:00 - 2022/05/11 20:30，一个半小时。
- 本次上机主要考察：结构体以及之前内容的综合应用
- 总共11道题，编号为A~K，部分题目由课件例题改编而来。
- **题目不一定按难度排序**，同学们可以按照通过人数答题；或者提前读完所有题目，按照自己擅长题目的情况答题。
- 各题分值分布为如下，总分103分，满分100分（超过100分按100分计算）：

序号	A	B	C	D	E	F	G	H	I	J	K
分值	20	20	20	10	10	10	3	3	3	2	2

- 请严肃比赛，严禁抄袭他人代码，课程组会在比赛结束后进行代码查重并给予警告。

总体规划与课件题目提供：谭火彬老师、陈高翔老师

组题：

更多题目作者：

## A 简单的OI启蒙

### 题目介绍

正如题目所说，这道题是JJJ的OI启蒙题。

陶陶家的院子里有一棵苹果树，每到秋天树上就会结出10个苹果。苹果成熟的时候，陶陶就会跑去摘苹果。陶陶有个30厘米高的板凳，当她不能直接用手摘到苹果的时候，就会踩到板凳上再试试。

现在已知10个苹果到地面的高度，以及陶陶把手伸直的时候能够达到的最大高度，请帮陶陶算一下她能够摘到的苹果的数目。假设她碰到苹果，苹果就会掉下来。

### 输入格式

第一行，10个整数(以厘米为单位)，分别表示10个苹果到地面的高度。

第二行，一个2位小数（以米为单位），表示陶陶把手伸直的时候能够达到的最大高度。

## 输出格式

一个整数，表示陶陶能够摘到的苹果的数目。

## 输入样例1

```
100 200 150 140 129 134 167 198 200 111
1.10
```

## 输出样例1

```
5
```

## 数据范围

对于100%的数据：

苹果距离地面的高度在 $[100, 200]$ 之间（单位：厘米），陶陶把手伸直的最大高度在 $[1.00, 1.20]$ 之间（单位：米）

## 后记

这是作者入坑C++后做的第二道题（第一道是a+b）

Porter: JJJ.

---

# B 求前导零的个数

## 题目描述

cbj一度沉迷于玩ARM嵌入式芯片。

所谓的前导零是一个变量从最高位开始的连续零的个数。

在ARM汇编指令集中有一条 `clz` 指令实现计算前导零的个数，但是**cbj**的计算机没有这个功能,需要你帮忙实现。

例如32位的16进制整数 `0xa9` (十进制的169)的二进制形式为 `0000 0000 0000 0000 0000 0000 1010 1001`，即 $bit_0, 3, 5, 7$ 为1，那么前导零有 $31 - 7 = 24$ 个。

## 输入

一个 `unsigned int` 范围内的整数 $a$ 。

注意 `0` 我们认为0有32个前导零。

## 输出

输出一行一个整数，为  $a$  的前导零的个数。

## 输入样例

169

## 输出样例

24

*author:cbj*

# C 简单结构体-给线段排排序

## 题目描述

现在有一堆在一维数轴上的线段，你能把他们捋清楚吗？

具体来说，有  $n$  条线段，线段的左端坐标为  $L$ ，右端坐标为  $R$ （保证  $L < R$ ）。具体的排序要求如下：

- 在左端坐标不一样的时候，按照左端坐标从小到大排序；

- 在左端坐标相等的时候，按照右端坐标从小到大排序。

## 输入格式

输入共  $n + 1$  行；

第一行为一个整数  $n$  ( $1 \leq n \leq 1000$ )，表示一共有  $n$  条需要排序的线段；

接下来  $n$  行，每行两个整数  $L, R$  ( $0 \leq L, R \leq 100$ )，表示每条线段的左端和右端坐标。

需要注意的是，可能会出现一模一样的线段，请将他们全部输出。

## 输出格式

一共输出  $n$  行，为  $n$  条线段从小到大排序之后的结果；

每一行两个整数，为每条线段的左端和右端坐标，以一个空格分隔。

## 输入样例

```
3
2 3
1 4
1 2
```

## 输出样例

```
1 2
1 4
2 3
```

## HINT

冒泡排序，选择排序，计数排序，`qsort` 快速排序均可通过本题。

如果觉得开两个数组太麻烦，不如试试结构体呢？

结构体的定义可以写成以下形式：

```
struct section {  
    int l, r;//左端点和右端点  
};  
struct section temp;//声明一个结构体变量  
struct section a[1010];//声明一个结构体数组
```

交换的时候类似int 的交换方式即可

```
//假设需要交换section结构体变量a和b  
struct section temp=a;  
a=b;  
b=temp;
```

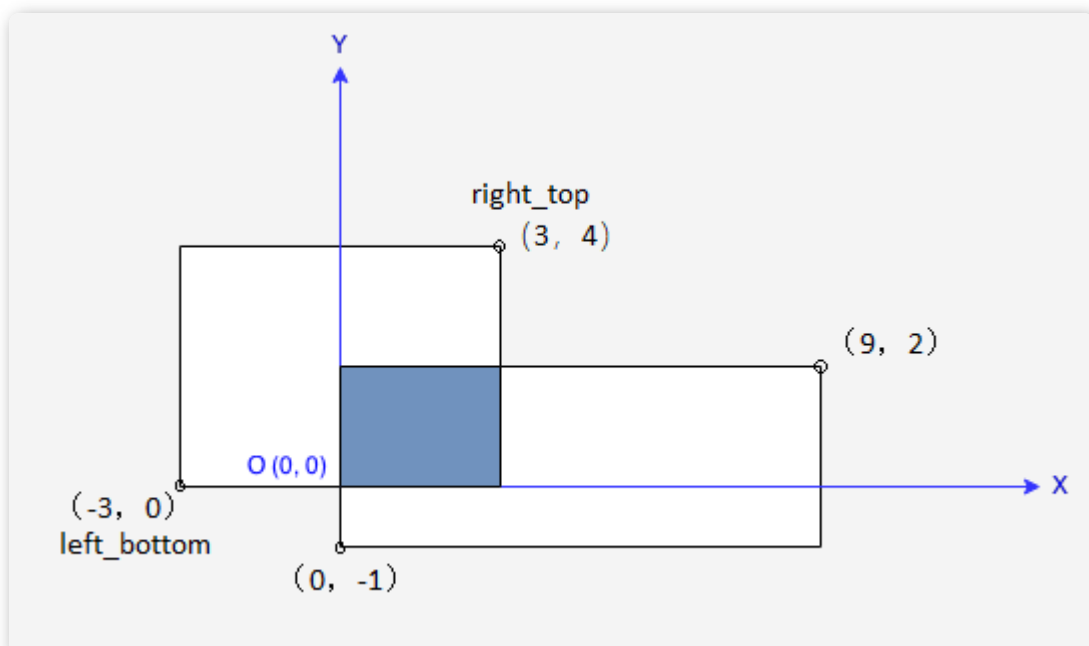
ps

题面来源于开花学长

## D 矩形重叠面积

### 题目描述

一个四边平行于坐标轴的二维矩形可由其左下角和右上角的坐标确定（如下图的矩形可由 $left\_bottom$ 和 $right\_top$ 两个点确定）。现给出两个矩形各自的左下角坐标和右上角坐标，求这两个矩形重叠部分的面积（即图中蓝色部分的面积）



## 输入

输入共两行

第一行为矩形1的左下角坐标和右上角坐标（共四个整数，按照左下角 $x$ 坐标，左下角 $y$ 坐标，右上角 $x$ 坐标，右上角 $y$ 坐标的顺序）

第二行为矩形2的左下角坐标和右上角坐标（顺序同上）

所有坐标均为 $int$ 型数据

## 输出

一个数，代表两矩形重叠部分的面积

## 输入样例

```
-3 0 3 4
0 -1 9 2
```

## 输出样例

```
6
```

## HINT

注意两矩形不相交的情况，此时重叠面积为0

另外，坐标这么多，不如试试用结构体吧

保证输出结果在 $long\ long$ 范围内

*Author:* 袁圆      *Porter:* wwh

---

# E 结构体小练习

## 题目描述

学生信息系统通常至少包含学生的姓名、学号、性别和联系方式等基本信息，对基本信息的维护是必不可少的，然而爱吃猪脚的猪脚去摸鱼了，只好请你维护一下学生信息系统了。

## 输入

共  $n + m + 1$  行。

第一行，两个正整数  $n, m$ ，表示初始共有  $n$  个学生，有  $m$  次操作。

接下来  $n$  行，每行由字符串  $s_1$ 、数字  $n_1$ 、字符串  $s_2$ 、数字  $n_2$  组成，分别表示学生的姓名、学号、性别和联系方式。

接下来  $m$  行，每行至少包含一个整数  $a$ ，具体规则如下：

- 若  $a = 1$ ，则表示在末尾增加一名学生，该行还包括字符串  $s_1$ 、数字  $n_1$ 、字符串  $s_2$ 、数字  $n_2$ ，分别表示新增学生的姓名、学号、性别和联系方式。
- 若  $a = 2$ ，则表示修改一名学生的联系方式，该行还包括数字  $n_1$  和数字  $n_2$ ，分别表示需要修改信息的学生的学号和修改后的联系方式。

## 输出

输出若干行，为维护后的学生信息，每行由字符串  $s_1$ 、数字  $n_1$ 、字符串  $s_2$ 、数字  $n_2$  组成，分别表示学生的姓名、学号、性别和联系方式。

输出顺序和输入一致。

## 输入样例

```
3 2
name1 12345678 male 13100000000
name2 87654321 female 13100000001
name3 11111111 male 13100000002
1 name4 22222222 female 13100000003
2 12345678 13188888888
```

## 输出样例

```
name1 12345678 male 13188888888
name2 87654321 female 13100000001
name3 11111111 male 13100000002
name4 22222222 female 13100000003
```

## 数据范围

$0 < n, m \leq 50$

$0 < \text{len}(s_1) \leq 50$ ，包含大小写字母和数字

保证  $n_1$  是 8 位整数且唯一

$s_2 \in \{male, female\}$

保证  $n_2$  是 11 位整数且唯一

## Hint

要不试试结构体？

PS：结构体大练习即将上线，敬请期待~

AUTHOR：爱吃猪脚的猪脚

# F 宋老师的名次预测5.0

## 题目描述

是的，我们是有一个5.0。

给出  $n$  个预测者的姓名、组别以及对 50 名选手的预测，请你根据猜对的人数对  $n$  个预测者进行排序并输出；

排序的规则为：首先按猜对的人数**从大到小**，猜对人数相同的情况下按组别**从小到大**，组别也相同的情况下按**姓名字典序从小到大**进行排序。

## 输入格式



输入共  $n + 1$  行；

第一行一个正整数  $n$ ，表示有  $n$  个预测者；

接下来  $n$  行，每行的格式为 [组别][空格][姓名][空格][对50名选手的预测]。

## 输出格式

输出共  $n$  行，为  $n$  个预测者按上述规则排序后的结果

每一行输出的格式为 [姓名][空格][组别][空格][猜对的人数]

## 输入样例

```
4
2 BodanChen 2 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50
1 HHH 2 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
46 47 48 49 50
1 CCC 2 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
46 47 48 49 50
2 SongYou 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
45 46 47 48 49 50
```

## 输出样例

```
SongYou 2 50
CCC 1 48
HHH 1 48
BodanChen 2 48
```

## 样例解释

预测者SongYou猜对了50个，另外三位都只猜对了48个，所以SongYou排第一位；在剩下三位中，CCC和HHH的组别比BodanChen小，所以BodanChen排最后一位；CCC和HHH中，猜对的人数和组别都相同，所以按照姓名的字典序从小到大排序，CCC排在前面，故输出如上。

## 数据范围

$$1 \leq n \leq 10^5$$

姓名只包含字母大小写，长度不超过 20，且互不相同

组别是 1 — 50 之间的正整数

## HINT

推荐使用 结构体+qsort快速排序

# G 图图玩转随机数

## 题目背景

图图完成了 *LJF* 算可靠度和图图的红包烦恼这两道题，现在已经把 `rand` 函数玩得很溜，不过他很好奇 `rand` 函数内部是如何实现的，这是他找到的资料：

随机数在计算机领域应用很广，抢红包数额、游戏抽卡牌、音乐随机播放等等都用到了随机数，计算机中通常实现的是伪随机数列。



**伪随机数**，是使用一个确定性的算法计算出来的似乎是随机的数序，因此伪随机数实际上并不随机

线性同余法就是其中一种比较简单的算法，C语言中的 `rand` 函数产生伪随机数实际上就采用了“线性同余法”：

$$X(n+1) = (a \times X(n) + c) \% m$$

其中参数  $a, c, m$  为常量，引入概念 *seed*（种子），它会作为  $X(0)$  被代入上式中，之后每次调用 `rand` 函数都会用上一次产生的随机值来生成新的随机值。

可以看出实际上用 `rand` 函数生成的是一个**递推**的序列，一切值都来源于最初的 *seed*，所以当初始的 *seed* 取一样的时候，得到的序列都相同；且产生的数列存在一个周期  $T$ ，满足  $T \leq m$ 。

## 题目描述

给出”线性同余“公式中的  $a, c, m$  以及种子  $X(0)$  的值，输出数列的周期  $T$ 。

## 输入格式

多组数据（不超过 100 组），每组一行，包括四个以空格分隔的正整数，分别表示  $a, c, m, X(0)$ 。

## 输出格式

对于每组输入，输出一行。

在参数  $a, c, m, X(0)$  的组合下，数列从某一项开始存在周期  $T$ 。如果通过观测数列前  $(10^4 + 1)$  项即能得出该周期  $T$ ，则以 `T = %d` 格式输出，否则输出 `T > 10^4`

## 输入样例

```
1 3 4 1
20 1 60000 1
1 1 10000 9999
214013 2531011 4294967296 1
```

## 输出样例

```
T = 4
T = 2
T = 10000
T > 10^4
```

## 样例解释

第一组数据， $X(n+1) = (X(n) + 3) \% 4$ ，产生的数列为 1-0-3-2-1-0-3-2.....周期 `T = 4`；

第二组数据， $X(n+1) = (20 \times X(n) + 1) \% 60000$ ，产生的数列为 1-21-421-8421-48421-8421-48421.....从第 4 项开始呈现周期性，周期 `T = 2`；

第三组数据， $X(n+1) = (X(n) + 1) \% 10000$ ，数列第  $(10^4 + 1)$  项等于数列第 1 项，即  $X(10000) = X(0)$ ，得到周期  $T = 10000$ ；

第四组数据周期大于  $10^4$ ，输出  $T > 10^4$

## 数据范围

$$0 < a, c, X(0) < 2^{31}$$

$$0 < m \leq 2^{32}$$

$$X(0) < m$$

## HINT

判断数组中是否存在重复数字，常用方法有两重循环查找、排序、利用数组下标，请你根据题目数据特点，选择合适方法来解决这个问题。

## 了解更多

[【柴知道】你在氪金游戏里抽的SSR是「真随机」吗？](#)

Author: wqh

# H 图图安排比赛

## 题目描述

番豆幼儿园将举行一场亲子运动会，健康哥哥邀请图图帮忙安排比赛。

比赛共有  $n$  支队伍，健康哥哥计划进行**单循环赛**，即所有参加比赛的队伍均能相遇一次，现介绍“逆时针轮转”编排单循环赛的方法。

假设  $n$  为偶数，则把参赛队平均分成左、右各一半，左一半号数由序号 1 开始依次自上向下排，右一半号数按顺序依次自下向上排，然后用横线相连，即构成比赛的第一轮。如  $n = 6$  时第一轮比赛编排如下：

Round 1

1-6

Round 1
2-5
3-4

从第二轮比赛开始，固定左上角的数字，其余数字按逆时针方向移动一个位置，以后各轮比赛秩序以此类推， $n = 6$  时的比赛各轮编排方式如下表：

Round 1	Round 2	Round 3	Round 4	Round 5
1-6	1-5	1-4	1-3	1-2
2-5	6-4	5-3	4-2	3-6
3-4	2-3	6-2	5-6	4-5

经过五轮比赛，所有参赛队伍均与其他队伍相遇一次。

若  $n$  为奇数，则在第一轮右上角补 0，与 0 相遇的队伍轮空一次，之后的比赛编排同以上规则。 $n = 7$  时的比赛各轮编排方式如下表：

Round 1	Round 2	Round 3	Round 4	Round 5	Round 6	Round 7
1-0	1-7	1-6	1-5	1-4	1-3	1-2
2-7	0-6	7-5	6-4	5-3	4-2	3-0
3-6	2-5	0-4	7-3	6-2	5-0	4-7
4-5	3-4	2-3	0-2	7-0	6-7	5-6

经过七轮比赛，所有参赛队伍均与其他队伍相遇一次。

可以发现，当参赛队为奇数时，比赛轮数  $round = n$ ，参赛队为偶数时，比赛轮数  $round = n - 1$ 。

现给定比赛队伍数  $n$ ，输出按如上规则编排的比赛表。

## 输入格式

一行，一个正整数  $n$  ( $2 < n < 10$ )，表示参加比赛的队伍数。

## 输出格式

输出 Round  $i$  表示第  $i$  轮比赛，接下来输出该轮比赛的编排表（本轮轮空的队伍不输出），球队编号之间用连字符 - 隔开。具体格式参照样例输出。

## 输入样例

7

## 输出样例

Round 1

2-7

3-6

4-5

Round 2

1-7

2-5

3-4

Round 3

1-6

7-5

2-3

Round 4

1-5

6-4

7-3

Round 5

1-4

5-3

6-2

Round 6

1-3

4-2

6-7

Round 7

1-2

4-7

5-6

*Author: wqh*

---

# I 简单的网页生成

## 题目描述

zym搭建了一个简单的动态网页。由于网页数据来自数据库，所以zym把动态的数据用 `{{ VAR }}` 的格式标记了出来，其中 `VAR` 是变量。

在生成具体网页的时候 `{{ VAR }}` 会被变量 `VAR` 的值代替。比如当变量定义为 `name="zym"` 时，`{{ name }}` 会变成 `zym`。现在给出包含变量的网页源代码和变量定义，请你编写一个程序输出将变量定义替换成变量值后的网页源代码。

其中关于变量的规则如下：

- 变量由大小写字母、数字和下划线 `_` 构成，第一个字符不是数字。且变量名对大小写敏感。
- 变量的值是字符串（可包含除了 `\` 以外任意可打印 ASCII 字符）。
- 变量名和变量值的长度不超过255。
- 如果标记中的变量名没有定义，则将标记从模板中删除。
- 标记没有递归生成。即使变量的值包括形如 `{{ VAR }}` 的内容，也不做进一步替换。
- 变量定义的格式为 `变量名 + 若干个空格 + " + 变量值 + "`
- 标记的格式为 `{{ + 空格 + 变量名 + 空格 }}`

## 输入格式

输入的第一行是两个整数 $m$ 和 $n$ 。 $m$ 表示包含变量的网页源代码行数， $n$ 表示需要进行替换的变量个数。

接下来 $m$ 行，每行为一个字符串，表示网页源代码。

接下来 $n$ 行，每行表示一个变量名和变量值，中间用若干个空格分开。 $n$ 行定义的变量不会重复，关于变量名和变量值的格式见问题描述。

## 输出

输出若干行，表示进行变量替换后的网页源代码。

## 输入样例

```
11 2
<!DOCTYPE html>
<html>
<head>
```

```
<title>User {{ name }}</title>
</head>
<body>
<h1>{{ name }}</h1>
<p>Email: <a href="mailto:{{ email }}">{{ email }}</a></p>
<p>Tel:{{ tel }}</p>
</body>
</html>
name      "zym"
email     "Anyakawaii@gmail.com"
```

## 输出样例

```
<!DOCTYPE html>
<html>
<head>
<title>User zym</title>
</head>
<body>
<h1>zym</h1>
<p>Email: <a
href="mailto:Anyakawaii@gmail.com">Anyakawaii@gmail.com</a></p>
<p>Tel:</p>
</body>
</html>
```

## 样例解释

第四行和第七行的 `{{ name }}` 替换成变量值 `zym`

第八行的两个 `{{ email }}` 替换成变量值 `Anyakawaii@gmail.com`

第九行的 `{{ tel }}` 由于没有定义，删除。

## 数据范围

$$0 \leq m \leq 255$$

$$0 \leq n \leq 255$$

输入的源代码以及替换后的源代码每行不超过300个字符

源代码中所有的 `{{` 都是一个合法的标记的起始部分



输入的所有变量名称不相同

## HINTS

1. 注意看具体格式的定义！
2. 其实是很简单的字符串替换问题~
3. 测评机上换行符是 `\r\n`。

*AUTHOR : zym*

# J 命运的波导

## 题目描述

“无论何时，我们的心都彼此相连！”（摘自小智忍蛙主题曲《XY&Z》主歌第二段）

小智（Satoshi）、甲贺忍蛙（Gekkouga）和路卡利欧（Lucario）三小只拥有同样的波导和牵绊，即使分属不同的地区，都可以在一瞬间同时相互感应到对方。它们三个一起设计了小智数列  $S_n$ 、忍蛙数列  $G_n$  和路卡数列  $L_n$ ：

$$S_n + G_n \sqrt[3]{2} + L_n \sqrt[3]{4} = (1 + \sqrt[3]{2} + \sqrt[3]{4})^n$$

三个数列都是整数列。由于线性无关性，整数列表示是唯一的。

这里给一个信息：数字  $1 + \sqrt[3]{2} + \sqrt[3]{4}$  比较特殊，它的倒数也能写成同样的形式。

/\* 在《代数数论》中，只有极少数的数拥有此性质，本题之外无需了解 \*/

$$(-1 + \sqrt[3]{2})(1 + \sqrt[3]{2} + \sqrt[3]{4}) = 1$$

因为小智和蛙狗三只忙着练巨大波导弹去了，于是打算让你来帮忙计算这个数列的各项是多少。为了方便，输出各项对 998244353 取模后的结果。

你可能不容易看懂上面的内容。这里举个例子：

$$(1 + \sqrt[3]{2} + \sqrt[3]{4})^{-2} = 1 - 2\sqrt[3]{2} + \sqrt[3]{4}$$

于是就有：

$$S_{-2} = 1 \quad G_{-2} = 998244351 \quad L_{-2} = 1$$

为了更加直白生动地讲清楚数列的结构，这里给一些初值，方便直观的了解。

$n$	-4	-3	-2	-1	0	1	2	3	4
$S_n$	-7	1	1	-1	1	1	5	19	73
$G_n$	-2	3	-2	1	0	1	4	15	58
$L_n$	6	-3	1	0	0	1	3	12	46

上述初值在输出时需要模 998244353。

为了快速计算对应下标的波导数列，这里提供一个快速幂的板子：

```
struct SGL FastPower(struct SGL base,long long exponent)
{
    struct SGL power;
    power.S=1;
    power.G=0;
    power.L=0;
    for(;exponent!=0;exponent>>=1)
    {
        if((exponent&1)==1)
        {
            power=Mult(power,base);
        }
        base=Mult(base,base);
    }
    return power;
}
```

当然，这个板子只支持正数幂的部分。板子中 *Mult* 表示结构体的乘积函数，小智希望您自己来设计。

## 输入

一个数，表示下标自变量  $n$ ，范围在 *int* 范围内，可正可负。

## 输出

对于每组数据，输出三个数，中间各有一个空格。先输出小智数列  $S_n$ ，再输出忍蛙数列  $G_n$ ，最后输出路卡数列  $L_n$ 。

## 输入样例

```
-2
```

## 输出样例

```
1 998244351 1
```

## Hint

本题仿照复数结构体，实现一种伪面向对象的编程。

举个例子：定义一个结构体叫复数，封装了两个成员实部和虚部，那么就需要写一些与该结构体紧密相关的函数，比如乘法函数、幂运算函数等等。这些函数的共同点，都是针对复数这种结构体类型的操作，而不是混合其他类型的操作。于是在程序的剩下部分，例如主体部分，只需要调用这些“专业”的函数，就能完成所有与复数结构体类型相关的操作。

*课外知识1：在面向对象里面，封装好的结构类型称为“类”，类的成员称为“属性”，与一个类密切相关的函数称为类的“方法”。所谓的“对象”，就是类的一个实例，即具体的一块内存。面向对象程序内部所有的操作都由对象的方法进行。*

*课外知识2：C99当中有<complex.h>头文件，以类型而不是结构体的方式实现复数操作；而C++98则借助头文件定义了复数对象和相应操作。本题不需要使用复数，而是仿照复数操作进行实现。*

本题改编自21年《兔狗贴贴》。五一赛T题《计算连分数》的思路与本题类似。



Author: 呱呱泡蛙

## K zhnの超级数列

### 题目描述

给定  $n$ ，求斐波那契数列第  $n$  项，结果对  $10^9 + 7$  取模。

定义前几项以及递推式子如下：

- $f(1) = 1$
- $f(2) = 1$
- $f(n) = f(n - 1) + f(n - 2)$

### 输入

一个正整数  $n$

### 输出

输出  $f(n)$  对  $10^9 + 7$  取模的值

## 输入样例

3

## 输出样例

2

对于40%的数据， $1 \leq n \leq 10^5$

对于100%的数据， $1 \leq n \leq 10^{18}$

*author:魔法少女zhn*

---