

# A. 来输出字符串吧！

时间限制：1000ms 内存限制：65536kb

磨刀不误砍柴工，请先仔细阅读比赛简介，了解比赛规则后再开始做题。

## 题目介绍

这里有一个字符串：

`"%o'\n/'o%"`

*JJJ*希望你能在计算机上把它输出出来。

## 输入样例

无。

## 输出样例

```
1 | "%o'\n/'o%"
```

## HINT

参考课件P2 c2-0

在C语言中，有一些字符不能够直接使用 `printf()` 输出，我们需要利用转义符来将其输出。

输出反斜杠 `\`，应写成 `printf("\\");`

输出双引号 `"`，应写成 `printf("\"");`

输出单引号 `'`，应写成 `printf("'");`

输出百分号 `%`，应写成 `printf("%%");`

所以想要输出这个字符串，想想应该怎么写

#

Author: *///*

# B. 字符也是数

时间限制：1000ms 内存限制：65536kb

## 题目描述

刚刚接触C语言的同学们都见过字符在屏幕上被打印出来的样子，那么字符是以什么形式存储在计算机中的呢？我们常见的字符都是以[ASCII码](#)——一种数字的形式储存在计算机中。比如字母 `b` 是98，符号 `!` 是33，而空格  是32。

随着计算机的普及和发展，有越来越多的字符编码系统出现，比如Unicode等，不过这些都是后话了...

## 输入

一行，一个可见字符。

## 输出

输出 `%d is the ASCII code for %c.`，其中 `%c` 代表输入的字符，`%d` 是输入字符的ascii码值。

## 输入样例1

```
1 | b
```

## 输出样例1

```
1 | 98 is the ASCII code for b.
```

## 输入样例2

```
1 | 
```

## 输出样例2

```
1 | 32 is the ASCII code for  .
```

## 样例2说明

其实输入了一个空格哦，不信你可以拿光标选中一下试试

## 关于可见字符

可见字符，也叫打印字符，是指可以被直接打印出来的字符，在ASCII码表中，从32到126共95个是可见字符。所以，空格也是可见字符。

## HINT

参考课件P2 c2-1

试试这段代码吧。

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char c;
6      scanf("%c", &c);
7      printf("%d", c);
8
9      return 0;
10 }
```

*Author: Lucien Li*

# C. 一元二次方程 真 便宜

时间限制：1000ms 内存限制：65536kb

## 题目介绍

小明同学有很多一元二次方程，你能帮他求解这些方程么？

方程的形式为  $ax^2 + bx + c = 0$  其中  $x$  为未知数， $a$ 、 $b$ 、 $c$  为参数。

## 输入格式

一行三个整数，分别表示参数  $a$ 、 $b$ 、 $c$

## 输出格式

若方程无解,输出 `no real roots`

如果有根，输出一行两个浮点数，以一个空格分隔，表示方程的两个根，保留到小数点后6位；

值小的在前，值大的在后，如果是重根，输出两个一样的数就好了

## 输入样例1

```
1 | 1 2 3
```

## 输出样例1

```
1 | no real roots
```

## 输入样例2

```
1 | 1 5 3
```

## 输出样例2

```
1 | -4.302776 -0.697224
```

## 数据范围

$$-10 \leq a, b, c \leq 10$$

$$a, b, c \neq 0$$

$$a, b, c \in \mathbb{Z}$$

## HINTS

参考课件P2 c2-6

①  $\Delta = \sqrt{b^2 - 4ac}$ 。

② 两个变量r1和r2求小值，代码为：

```
1  if(r1<r2)
2      min=r1;
3  else
4      min=r2;
```

③开根号可以利用 `math.h` 中的 `sqrt()` 函数。

[C 库函数- sqrt\(\)](#)

④下面给出保留两位小数和三位小数的示例代码

```
1  double a=1.23456,b=1.23456;
2  printf("%.2f",a)//保留两位小数
3  printf("%.3f",b)//保留三位小数
```

*AUTHOR:ljf*

# D. 火仙草

时间限制：200ms 内存限制：65536kb

## 题目描述

对于一个四位数，如果它**满足下列条件之一**，则称它为火仙草数：

- 前两位<sup>2</sup> + 后两位<sup>2</sup> = 原数
- 千位<sup>4</sup> + 百位<sup>4</sup> + 十位<sup>4</sup> + 个位<sup>4</sup> = 原数
- 原数是 666 或 888 的倍数

例如 8208 是火仙草数，因为  $8^4 + 2^4 + 0^4 + 8^4 = 4096 + 16 + 0 + 4096 = 8208$ 。

现在，输入一个四位数  $n$ ，请你判断  $[1000, n]$  内有多少个火仙草数。

## 输入

共一行，一个四位整数  $n$ ，含义如题所述。

## 输出

共一行，一个整数，表示范围内有多少个火仙草数。

## 输入样例

1 | 1500

## 输出样例

1 | 2

## 样例解释

$[1000, 1500]$  范围内， $1233 = 12^2 + 33^2 = 144 + 1089$  和  $1332 = 2 \times 666$  是火仙草数，因此输出 2。

## Hint

参考课件P2 c2-9

Author：爱吃猪脚的猪脚

# E. 简单后缀运算

时间限制：1000ms 内存限制：65536kb

## 题目描述

**后缀表达式**，又称**逆波兰式**，指的是不包含括号，运算符放在两个运算对象的后面，所有的计算按运算符出现的顺序，严格从左向右进行（不再考虑运算符的优先规则）的表达式。

比如，我们常用的中缀表达式  $3 + 5$  写成后缀表达式的形式就是  $3\ 5\ +$ ， $1 + 2 * 3$  的后缀形式是  $1\ 2\ 3\ *\ +$

下面，我们给出一些只有一个**运算符**的简单后缀表达式，请你输出其结果

## 输入格式

多组数据输入，以文件结束符 EOF 为结尾；

每组数据一行，格式为  $\%d\ \%d\ \%c$ ，其中  $\%d$  是**整数**运算对象， $\%c$  是运算符， $\%c \in \{+, -, *, /, \%, \}$

$/$  是整除运算， $\%$  是求模运算，均按C语言运算规则计算；

## 输出格式

每组数据输出一行，表示该后缀表达式运算的结果；

当运算为整除 or 求模时，如果除数为 0，输出 `Runtime Error (SIGFPE)`

## 输入样例

```
1 3 5 +
2 6 7 *
3 0 8 /
4 8 0 %
5 1 1111 -
```

## 输出样例

```
1 8
2 42
3 0
4 Runtime Error (SIGFPE)
5 -1110
```

## 样例解释

样例中的5组数据的中缀形式分别为  $3 + 5$ ， $6 * 7$ ， $0/8$ ， $8\%0$ ， $1 - 1111$ ，故输出如上

## 数据范围

所有输入的运算对象以及运算结果都在 `int` 范围内

## HINT

```
1 int a,b;  
2 char ch;  
3 // 考虑下面两个语句的区别  
4 scanf("%d%d%c",&a,&b,&ch);  
5 scanf("%d%d %c",&a,&b,&ch);
```

## PS

中缀表达式更符合人类的计算思维，但是对于计算机来说，后缀表达式显然更符合计算的逻辑，对于一个后缀表达式，只需要从左往右扫描一遍，如果是运算对象的话就先暂时存储起来，如果是运算符就将最新的两个运算对象进行运算，如此进行下去就可以很简单的完成整个表达式的计算；所以将中缀表达式转换成后缀表达式是一个聪明而重要的过程；

本题只涉及一个运算符，很简单地通过判断就能够进行计算，那么对于较复杂的中缀表达式，比如  $1+((2+3)\times 4)-5$ ，其后缀表达式又是多少呢？对于任意一个中缀表达式，怎么将其转换成后缀表达式呢？这需要用到一个叫 **栈** 的数据结构，在后续的相关课程中可能会进行学习，有兴趣的同学可以自行了解。

AUTHOR:陈博胆



# F. 字符变变变

时间限制：1000ms 内存限制：65536kb

## 题目描述

输入一段非空字符串，对其进行如下操作：

- 小写字母转换成相应的大写字母
- 大写字母转换成相应的小写字母
- 其他的字符不变

请输出转换后的字符串。

## 输入格式

一行，一个字符串，仅包含可见字符；

## 输出格式

一行，一个转换后的字符串；

## 输入样例

```
1 | HTTPS: //PASTEBIN.COM/NY3A0BEB
```

## 输出样例

```
1 | https: //pastebin.com/ny3a0beB
```

## 数据范围

字符串长度不超过 100

## HINT

参考课件P2 c2-8

AUTHOR: cbd

# G. 统计统计成绩

时间限制：1000ms 内存限制：65536kb

## 题目描述

给出  $n$  个同学的成绩，请你统计成绩**大于等于**平均分的同学的平均分。

## 输入

第一行一个整数  $n$ ，表示人数。

接下来一行， $n$  个整数，分别表示  $n$  个同学的成绩。

## 输出

一个小数，代表成绩**大于等于**平均分的同学的平均分。请**保留两位小数**。

## 输入样例

```
1 | 10
2 | 89 76 20 66 59 92 100 70 85 88
```

## 输出样例

```
1 | 88.33
```

## 样例解释

这10位同学的平均分为74.50分，其中在平均分之上的有：89，76，92，100，85，88，这6位同学的平均分为88.33分。

## 数据范围

$$1 \leq n \leq 1000$$

$$0 \leq scores \leq 100$$

## HINT

本题你可能需要用到 `数组` 的相关知识，参考课件P2 c2-11-1

# H. 甲贺忍蛙的名次预测

时间限制：1000ms 内存限制：65536kb

## 题目描述

鉴于训练家小智即将升入宝可梦世界锦标赛的八大师，甲贺忍蛙（Gekkouga）在赛前预测八位选手，在比赛中会按顺序分别获得第1名到第8名。

## 输入

多组数据输入，每组数据一行，一行八个整数，依次是八位选手的实际名次的一种可能性。

## 输出

对于每组数据：

如果对于这种可能性，至少有5个人的名次被甲贺忍蛙猜对，那么输出一行“Kouga!”。

如果对于这种可能性，最多有4个人的名次被甲贺忍蛙猜对，那么输出一行“Gek...”。

## 输入样例

```
1 1 8 3 6 5 4 7 2
2 1 2 3 5 4 6 7 8
```

## 输出样例

```
1 Gek...
2 Kouga!
```

## 样例解释

第一组数据与甲贺忍蛙的预测序列1 2 3 4 5 6 7 8之间只有4个一致，因此程序输出一行“Gek...”。

第二组数据与甲贺忍蛙的预测序列1 2 3 4 5 6 7 8之间已经有6个一致，因此程序输出一行“Kouga!”。

## Hint

题目搬运工跑去看动画回归的甲贺忍蛙去了。同学们可以参考课件c2-10。



课件题目作者：宋老师

题目搬运工：想成为甲贺忍蛙的呱呱泡蛙

# I. 时刻2.0

时间限制：1000ms 内存限制：65536kb

## 题目描述

这是一道没那么简单的判断题。请判断两个时刻的早晚关系。

## 输入

多组数据输入，共  $2n + 1$  行。

第一行，一个正整数  $n$ ，表示有  $n$  组数据。

接下来每 2 行为一组数据，共  $2n$  行。

这两行每行 1 个时刻，格式为 `aa hh:mm`，为相同时区的同一天的 12 小时制的时刻。

## 输出

每组数据输出一行，共输出  $n$  行，对于每组数据：

如果第一行时刻晚于第二行，输出 `1`。

如果第二行时刻晚于第一行，输出 `2`。

如果两个时刻相同，输出 `Same Time`。

## 输入样例

```
1 3
2 AM 12:12
3 AM 08:13
4 PM 12:12
5 AM 08:13
6 AM 12:12
7 AM 12:12
```

## 输出样例

```
1 2
2 1
3 Same Time
```

## 样例解释

12小时制的中文含义与英文表达对应如下

中文	英文表达
午夜 12:00	AM 12:00
凌晨 1:00	AM 1:00
...	...
上午 11:00	AM 11:00
中午 12:00	PM 12:00
下午 1:00	PM 1:00
...	...
深夜 11:00	PM 11:00

## 数据范围

---

12 小时制时间采用如下格式：

aa 为 AM 或 PM 标记，均为大写。

hh 为小时数 1-12，即 PM 11:59 之后为 AM 12:00。

mm 为分钟数 0-59。

## Hint

---

读入第二行数据之前请想一想，怎么处理第一行行末的换行符。

参考课件P2 c2-2r

Author: 爱吃猪脚的猪脚

# J. 科学计数法!

时间限制：1000ms 内存限制：65536kb

## 问题描述

众所周知，基物实验是大部分北航学子都要修的一门课程，在处理实验数据时，有时候会遇到值很大或者很小的数据，在这个时候采用**科学计数法**是一种重要的实验处理方法；

下面请你将一个小数表示的浮点数，转换成科学计数法的形式吧！

## 输入格式

一行，一个小数表示的**正浮点数**；

输入的数据均为有效数据，即小数点后的末尾数字不为 0；小数点前若只有一位数字，可以为 0，否则小数点前最高位数字不为 0

## 输出格式

以科学计数法的形式输出数据，输出的数据由以下几个部分构成：

- 底数部分是一个小数或者整数，若为小数，则小数点前应只有一位大于 0 的数字，小数点后的末尾数字不能为 0；若为整数，则只有一位数字，不带小数点；
- 必须有小写字母 **e**；
- 指数部分是一个整数，若大于等于 0 则不带正号 **+**，小于 0 需要带负号 **-**，且如果指数不为 0 的话不能有前导零；

## 输入样例1

```
1 | 0.000023
```

## 输出样例1

```
1 | 2.3e-5
```

## 输入样例2

```
1 | 3.14159265358979323846
```

## 输出样例2

```
1 | 3.14159265358979323846e0
```

## 输入样例3

```
1 | 123456789098765432123456789098765432.1
```

## 输出样例4

## 数据范围

---

输入的浮点数字符个数不超过100

*AUTHOR: cbd*