

# Problem A. 佛像

## 题目描述

某只蒟蒻最近的ddl太多了，还总是记错内容和时间，请你为ta输出一尊佛像。



## 样例输出

```

      _ooOoo_
      o8888888o
      88" . "88
      (| -_- |)
      O\ = /O
      %____/'---'\____%
      . ' \ \ | | // ' .
      / \ \ | | : | | // \
      / _ \ \ | | -:- | | | - \
      | | \ \ | | - /// | |
      | \ | ' ' \---/' ' | |
      \ .- \ | | \ _ _ / - . /
      _ _ \ . ' / - - - \ _ _ \
      ."" ' < ` . _ _ \ < | > / _ _ . ' > '"" .
      | | : ` - \ \ . ; \ _ / ; . / - ` : | |
      \ \ `-. \ _ _ / _ _ / .-` / /
      %%===== `-. _ _ `-. _ _ \ _ _ / _ _ .-` _ _ .-` ===== %%
      `-----'

```

## Hint

单纯的复制粘贴可不行嗷~

# Problem B. 温度转换

## 题目描述

温度转换打表。

op为0时，将 $[L, R]$ 的摄氏温度转换成华氏温度输出。

op为1时，将 $[L, R]$ 的华氏温度转换成摄氏温度输出。

## 输入格式

第一行一个整数op（0或1）。

第二行两个0 – 100 之间的整数 $L, R, L \leq R$ 。

## 输出格式

输出 $R - L + 1$ 行，

每行为空格分隔的两种温度，转换后的温度保留两位小数。

## 输入样例

```
1
90 95
```

## 输出样例

```
90 32.22
91 32.78
92 33.33
93 33.89
94 34.44
95 35.00
```

## Hint

华氏度（ $F$ ）、摄氏度（ $C$ ）转换公式：

$$F = \frac{9}{5}C + 32$$
$$C = \frac{5}{9}(F - 32)$$

author XIAO7

## Problem C. 基础物理实验

有一门很麻烦的课叫基础物理实验，它的积分规则如下：

- 每一门实验都有一个七位数的编号，**最后一位是完成这门实验后可以获得的积分**。如完成实验1010113可以获得3分。
- 按时到场但没有完成实验，不会获得积分，但也并不会倒扣积分。
- 迟到20分钟以上不仅不能获得积分，还会倒扣2分。（最多扣成0分）
- 还有很多很麻烦的规则就不考虑了QAQ。

请你帮一只蒟蒻计算一下ta的基础物理实验课可以拿到多少积分QAQ。

## 输入格式

第一行一个整数  $n$  ( $1 \leq n \leq 100$ ) , 表示实验的数量。

接下来  $n$  行, 每行两个空格分隔的整数  $id$  和  $op$ 。

- $id$  为一个没有前导 0 的七位整数, 表示实验的编号。
- $op$  表示实验完成的状态, 其值只可能为 0 或 1 或 2。
  - $op = 0$  表示成功完成了实验;
  - $op = 1$  表示按时到场但没有完成实验;
  - $op = 2$  表示迟到了 20 分钟以上QAQ。

## 输出格式

一行, 若这只蒟蒻的最终积分  $\geq 37$ , 则输出最终积分。若  $< 37$  表示这只蒟蒻明年需要重修了, 输出

See you next year !

## 样例输入

```
7
1010113 0
1010212 0
1030113 0
1030211 0
1030312 1
1030412 1
1020114 2
```

## 样例输出

See you next year !

## 样例解释

这只蒟蒻最终只获得了可怜的 7 分, 需要明年重修QAQ。

# Problem D. 子串逆置\_PRO

## 题目描述

众所周知, 课件里说了很重要的东西真的很重要。

## 2. 指向同一数组的两个指针进行比较

[c7-8] 子串逆置。从标准输入上读入以空格分隔的字符串  $s$  和  $t$ ，将  $s$  中首次与  $t$  匹配的子串逆置后再输出  $s$ ，当  $s$  中无与  $t$  匹配的子串时直接输出  $s$ 。

输入样例：  
abc123defg 123  
abc123defg 234

输出样例：  
abc321defg  
abc123defg

```
#include<stdio.h>
#include<string.h>
void rev(char *, char *);
int main(){
    char str[BUFSIZ], substr[BUFSIZ], *p;
    scanf("%s%s", str, substr);
    if((p = strstr(str, substr)) != NULL)
        rev(p, p + strlen(substr)-1);
    puts(str);
    return 0;
}
```

```
void rev(char* first, char* last)
{
    int tmp;
    while(first < last)
    {
        tmp = *last;
        *last = *first;
        *first = tmp;
        first++, last--;
    }
}
```

这个程序非常实用，记下来，牢固掌握，灵活运用！

XIAO7把题面改了改，希望你将  $S$  中所有与  $T$  匹配的子串都逆置过后输出。

她相信你可以把ppt上的代码小小改一下，就AC这道题目。

输入输出格式如图。

## 输入格式

一行两个空格分隔的字符串  $S$  和  $T$ ， $S$  和  $T$  只有大小写字母和数字组成， $S$ 、 $T$  的长度  $\leq 200$ 。

## 输出格式

一行，经过子串逆置变化后的字符串  $S$ 。

## 样例输入

```
windowsPowerShell ow
```

## 样例输出

```
windwosPwoerShell
```

## Hint

$S$  中与  $T$  匹配的字串互不相交，且不用考虑逆置后会构成新的字串的情况。

author XIAO7

# Problem E. Gcd-Expression

## 题目描述

正整数  $a$  和  $b$  的最大公约数可表示为

$$d = \gcd(a, b) = ax + by$$

即能被  $a$  和  $b$  线性表出的最小正整数，其中， $x$  和  $y$  是整数（可能为 0 或负数）。

输入  $a$  和  $b$ ，求出如上的表达式，见样例格式。

## 输入格式

一行，两个空格分隔的 `int` 范围内的整数  $a, b$ 。

## 输出格式

一行，`d = a*(x) + b*(y)`。

## 样例输入1

```
28 63
```

## 样例输出1

```
7 = 28*(-2) + 63*(1)
```

## 样例输入2

```
2029126176 1613441024
```

## 样例输出2

```
32 = 2029126176*(4726833) + 1613441024*(-5944649)
```

## Hint

伪代码如下：

```
EXTENDED-EUCLID( $a, b$ )
1  if  $b == 0$ 
2      return ( $a, 1, 0$ )
3  else ( $d', x', y'$ ) = EXTENDED-EUCLID( $b, a \bmod b$ )
4      ( $d, x, y$ ) = ( $d', y', x' - \lfloor a/b \rfloor y'$ )
5      return ( $d, x, y$ )
```

以上递归函数每次会返回 3 个值  $(d, x, y)$ ，应该怎么实现这个效果呢，或许可以传三个指针参数进入函数，在函数中修改它们指向的值：

```
#include<stdio.h>
void Example(int *a,int *b,int *c)
{
    *a=1,*b=2,*c=3;
}
int main()
{
    int a=0,b=0,c=0;
    printf("%d %d %d\n",a,b,c);
    Example(&a,&b,&c);
    printf("%d %d %d\n",a,b,c);
}
```

这样就能实现 返回3个值 的效果~

```
void Exgcd(int a,int b,int *d,int *x,int *y)
{
    //body
}
```

## Problem F. 成绩处理

### 题目描述

一个班级内共有 $N$  ( $0 < N \leq 1000$ )位同学，请你根据他们的成绩为他们排序。在成绩相同时，按照他们的姓名字典序排序（与 `strcmp` 函数返回的比较结果一致）。

### 输入格式

共输入 $N + 1$ 行。

第一行一个正整数 $N$ ，表示该班同学数量。

第 $i + 1$  ( $1 \leq i \leq N$ )行，有一个只含小写字母的字符串（长度不超过10，可能存在重名），表示第 $i$ 名同学的姓名，之后输入一个空格，之后输入一个整数表示第 $i$ 名同学的成绩 ( $\leq 100$ )。

### 输出格式

共输出 $N$ 行。

第 $i$ 行先输出一个字符串，表示按照上述规则排序后的第 $i$ 名同学的姓名，之后输出一个空格，之后输出一个整数表示这位同学的成绩。

### 样例输入1

```
6
tim 50
stywang 0
ohsheep 100
ghy 0
monica 100
john 50
```

### 样例输出1

```
monica 100
ohsheep 100
john 50
tim 50
ghy 0
stywang 0
```

## 样例输入2

```
5
a 0
b 0
aa 0
b 2
a 2
```

## 样例输出2

```
a 2
b 2
a 0
aa 0
b 0
```

## Hint

- 下面提供一个使用 `qsort` 函数的思路（当然用冒泡排序也是可以的~）：

用全局变量 `Score` 数组和 `Name` 数组按输入顺序存储成绩和姓名，用数组 `index[i]` 表示排序后排名第  $i$  的学生的数据输入的次序（也是 `Score` 数组和 `Name` 数组中的对应下标）。定义 `qsort` 函数中的 `Cmp` 函数，根据 `Score` 和 `Name` 数组的值，对 `index` 数组进行排序。

```
int Cmp(const void *pa, const void *pb) // 传入指针
{
    int a = *((int *)pa);
    int b = *((int *)pb);              // 用取内容运算符获取指针指向的数据

    if (Score[a] > Score[b])           // 比较成绩
    {
        return -1;                    // 返回-1表示前者排在前
    }
    return 1;                          // 返回1表示后者排在前
}
```

这是一个仅根据成绩对 `index` 进行排序的 `Cmp` 函数，你该如何修改使它支持分数相同时按照字典序排序呢？

- Author: StyWang

## Problem G. 划分测试集

## 题目描述

WXZ 正在训练一个图像分类卷积神经网络，他现在收集了很多的图片，每一张图片都属于一个类别。

WXZ 需要把这些图片分成两个数据集：训练集与测试集。其中，训练集中图片的数量需要是测试集的  $n-1$  倍。为了验证卷积神经网络分类的准确性，需要在测试集中囊括种类尽可能多的图片。

现在，请你帮他计算一下，在测试集中最多可以囊括多少个类别的图片？

## 输入格式

第一行两个数字  $n$   $k$ ， $n-1$  为训练集中图片数量与测试集中图片数量之比， $k$  为收集的图片个数，保证  $n$  可以整除  $k$ 。 $k \in [n, 1000000]$ ， $n \geq 2$ 。

第二行有  $k$  个整数  $a_i$ ， $i \in [1, k]$ ， $a_i$  代表第  $i$  张图片所属的类别。 $a_i \in [-100000, 100000]$ 。

## 输出格式

一行一个整数，表示测试集中最多可以囊括多少个类别的图片。

## 样例输入1

```
3 9
1 1 1 2 2 2 3 3 3
```

## 样例输出1

```
3
```

## 样例输入2

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

## 样例输出2

```
2
```

## 样例解释

对样例1，测试集中照片有  $9/3=3$  张，由于一共有三类照片，每类照片有三张，因此测试集中可以囊括所有种类的照片。

对样例2，测试集中有  $4/2=2$  张照片，不管测试集中的照片类别是  $\{1, 2\}$  还是  $\{1, 3\}$  还是  $\{2, 3\}$ ，测试集中最多可以有两类照片。

## Hint

或许你可以统计一下照片的种类一共有多少种~

或许可以用一个数组来统计某个类别的图片（即  $a_i$  为某个特定值的图片）是否出现过，有多少张~

AUTHOR : ghy



# Problem H. 错误的幻方

## 题目描述

幻方是一种很神奇的矩阵，一个  $N$  阶幻方是将从 1 到  $N^2$  的自然数排成纵横各为  $N$  排的正方形，且每行、每列及两条对角线上的数字之和都相同。

有一只不会算数的蒟蒻得到了一个错误的  $N$  阶幻方，已知交换其中两个数字即可得到正确的  $N$  阶幻方，请找到这两个数字，并输出它们的行列位置，按照行数优先，列数其次从小到大输出。

## 输入格式

第一行一个整数  $N$  ( $1 \leq N \leq 10^3$ )，表示幻方的阶数，接下来  $N$  行每行  $N$  个数，表示上述的错误的幻方。

保证该矩阵有且仅有一组数字在交换后使其满足幻方矩阵。

## 输出格式

输出共 2 行，每行两个整数，代表需要交换的两个数字的行数和列数，行数和列数从第一行、第一列开始计算。

## 样例输入

```
3
8 1 6
3 5 9
4 7 2
```

## 样例输出

```
2 3
3 2
```

## 样例解释

交换原本矩阵中的第二行第三个数 9 和第三行第二个数 7 后得到矩阵：

```
8 1 6
3 5 7
4 9 2
```

上述矩阵每行每列以及两条对角线的和均为 15，是一个正确的幻方。

# Problem I. 很解压的字符串解压

## 题目描述

现给定一个经过压缩的字符串，请你对其进行解压缩输出。

对于连续的若干相同子串 `STRING`，我们可以将其压缩为：`D(STRING)`，表示括号内部的 `STRING` 正好重复 `D` 次（`D` 为正整数）。

- 输入的字符串中所有的数字只表示重复的次数，即解压之后的字符串中**不包含数字**
- 注意存在**多重压缩**的情况，详见样例

## 输入格式

一行，表示被压缩的字符串（只包含数字、字母、`(`和`)`）。

## 输出格式

一行，表示被解压之后的字符串（长度不超过10000）。

## 样例输入1

```
WA3(AC)
```

## 样例输出1

```
WAACACAC
```

## 样例输入2

```
PE3(TLE2(CE))
```

## 样例输出2

```
PETLECECETLECECETLECECE
```

## 样例输入3

```
12(WA)AC10(TLE)
```

## 样例输出3

```
WAWAWAWAWAWAWAWAWAWAACTLETLETLETLETLETLETLETLETLE
```

## Hint

```
while(*p){
    if(isalpha(*p)){
        //store the alpha
    }
    else if(isdigit(*p)){
        //update the number
    }
    else if(*p == '('){
        //recursion
    }
}
```

```
else if(*p == ' '){
    //end
}
p++;
}
```

AUTHOR: cxc

# Problem J. MIPS

## 题目描述

这是某只蒟蒻第一次出大模拟，以至于ta太过于害怕而把计组课上学的 MIPS 汇编语言给忘了个一干二净，所以希望你们能帮帮ta。

因为蒟蒻非常的弱，所以ta把 MIPS 语言弱化了，弱化后 MIPS 语言如下：

- MIPS 语言的每一行是一句指令，在未遇见跳转语句的时候顺序从上往下执行。
- MIPS 的程序结构为：**（一定只有一个数据声明部分和一个代码声明部分，且数据声明部分在上方）**

```
.data

... .. # 数据声明部分

.text

... .. # 代码声明部分
```

- 在数据声明部分的指令集如下，可以定义 int 类型的变量和数组。

语句	参数解释	功能
name: .word val	name 为一个有大小写字母、数字、下划线组成的字符串; val 为一个 int 类型的整数.	定义一个名字为 name、值为 val 的 int 类型变量。
name: .space SIZE	name 为一个有大小写字母、数字、下划线组成的字符串; SIZE 为一个 $\leq 100$ 的正整数.	定义一个名字为 name、长度为 $\frac{SIZE}{4}$ 的 int 类型一维数组。（也可以理解为声明了一段大小为 SIZE 的连续的内存空间）

- 在代码声明部分的指令集如下：

弱化版的 MIPS 语言中存在 8 个寄存器，你可以理解为 8 个指针或地址，记作 \$s0、\$s1、⋯、\$s7；

同时还存在三个特别的寄存器（指针）\$a0、\$v0、\$zero。

注：寄存器（指针）\$zero 指向的地址里存的值恒为 0，题目也不会对其进行修改。

以下的指令集都是对以上的寄存器（指针）的操作，描述指令集时，使用 \$t0、\$t1、⋯代指以上 11 个寄存器。

赋值类语句

语句	参数解释	功能
li \$t0, val	\$t0 为任一寄存器(指针), val 为 int 型整数。	将 \$t0 指向的地址上保存的数的值赋值为 val。
move \$t0, \$t1	\$t0 和 \$t1 是任意两寄存器。	将 \$t1 赋值给 \$t0。

#### 操作地址类语句

语句	参数解释	功能
la \$t0, name	t0 为任一寄存器, name 为 <b>数据声明部分</b> 中定义的一个变量或数组。	使寄存器（指针）\$t0 指向名为 name 的变量或数组。
lw \$t0, delta(\$t1)	\$t0 和 \$t1 是任意两寄存器, \$t1 处储存了一个地址（指针）, delta 为一个整数（可能为整数、负数或 0）。	设 \$t1 处储存了一个地址p, 将p+delta处储存的值取出来储存在 \$t0 处。
sw \$t0, delta(\$t1)	\$t0 和 \$t1 是任意两寄存器, \$t1 处储存了一个地址（指针）, delta 为一个整数（可能为整数、负数或 0）。	设 \$t1 处储存了一个地址p, 将储存在 \$t0 处的值取出来储存在p+delta处。

#### 计算类语句

语句	参数解释	功能
add \$t0, \$t1, \$t2	\$t0、\$t1、\$t2 为任意三个寄存器。	将储存在 \$t1 和 \$t2 的值相加, 把和储存在 \$t0 处。
addi \$t0, delta	\$t0、\$t1 为任意两个寄存器, delta 为 int 类型的整数。	将储存在 \$t1 的值与 delta 相加, 把和储存在 \$t0 处。
mul \$t0, \$t1, \$t2	\$t0、\$t1、\$t2 为任意三个寄存器。	将储存在 \$t1 和 \$t2 的值相乘, 把积储存在 \$t0 处。

#### 关于标签和跳转

某一行作为一个标签当前仅当其形如 `label_name:`, `label_name` 为一个有大小写字母、数字、下划线组成的字符串, 表示标签的名字。（一定独占一行, 且是代码声明部分中唯一带冒号的语句）

跳转语句:

语句	参数解释	功能
j label_name	label_name 为一个标签的名字（可能在当前语句的上方或下方）	立即跳转到该标签所在的行。

#### 分支语句

语句	参数解释	功能
beq \$t0, \$t1, label	\$t1, \$t2 为任意两个寄存器，label 为一个标签的名字（可能在当前语句的上方或下方）	若 \$t1 和 \$t2 处的值相等则立刻跳转至 label 标签所在的行。
bne \$t0, \$t1, label	\$t1, \$t2 为任意两个寄存器，label 为一个标签的名字（可能在当前语句的上方或下方）	若 \$t1 和 \$t2 处的值不相等则立刻跳转至 label 标签所在的行。

syscall 语句（输出，结束程序）

\$v0	功能
当 \$v0 处储存的值为 1 时	输出 \$a0 处储存的 int 型整数。
当 \$v0 处储存的值为 11 时	将 \$a0 处储存的数的最低 8 位二进制 ascii 码对应的字符输出。
当 \$v0 处储存的值为 10 时	结束程序。

给出一段**没有输入**的弱化版 MIPS 代码（**保证代码合法**），请你输出它的运行结果。

## 输入格式

若干行，一个弱化版 MIPS 代码。保证代码合法且只会出现上述指令。

## 输出格式

若干行，输入的代码的输出结果。

## 样例输入1

```
.data
.text
li $s0, 2
li $s1, 3
mul $s2, $s1, $s0
move $a0, $s2
li $v0, 1
syscall
li $v0, 10
syscall
```

## 样例输出1

```
6
```

## 样例输入2

```

.data
.text
    li $s0, 0
    li $s1, 5
loop:
    beq $s0, $s1, Exit
    addi $a0, $s0, 65
    li $v0, 11
    syscall
    addi $s0, $s0, 1
    j loop
Exit:
    li $v0, 10
    syscall

```

## 样例输出2

ABCDE

## 样例输入3

```

.data
    val: .word 10
.text
    la $s0, val
    lw $a0, 0($s0)
    li $v0, 1
    syscall
Exit:
    li $v0, 10
    syscall

```

## 样例输出3

10

## 数据范围

指令行数  $\leq 100$ ，变量名、标签名长度  $\leq 10$ 。

## Hint

- 这不是一个考察字符串的题目，因此输入的代码格式都非常工整。
  - 如 `name: .word val` 和 `name: .space SIZE` 语句中：一定紧跟着 `name`，`name:`、`.word` 和 `val` 之间一定由空格隔开。
  - 逗号一定紧跟着其前面的参数，且后面一个有至少一个空格，例如输入语句可能是 `add $t0, $t1, $t2`、`add $t0, $t1, $t2` 等，但不可能是 `add $t0 , $t1 , $t2`。（其它指令以此类推）
  - 指令的行首和行末可能存在一些多余的空格，请忽略掉它们。

- 若寄存器里储存的值是一个地址，`add`、`addi`也可以对其进行运算，和C语言一样，地址可以加一个常数，但两个地址之间不能相加或相乘（当然数据中也不会出现这种非法情况）。  
注意对于地址 `int` 型地址 `p`，`p+1`（`addi p, p, 1`）的意思是将 `p` 往后移动**一个字节**（这和C是不一样的）。
- 再次强调一下**输入的代码一定合法**，比如非法的地址运算，调用空指针（未初始化的寄存器）、尝试用 `li $v0, 1` 和 `syscall` 输出一个地址（即这两句命令只能输出一个 `int` 型，地址虽可以看作16进制数，但仍是 `int*` 类型）、尝试修改 `$zero` 或 `$zero` 处储存的值，以及其它非法操作都不会在输入代码中出现。
- 以及最好不要偷懒，比如进行一些 `int` 和 `int*` 之间的强制转换，这样大概率会收获一个RE（出题人哭哭.jpg）。