第十四届重庆大学程序设计大赛

暨西南地区高校邀请赛

现 场 决 赛

The 14th CQU Programming Contest Final

The 14th CQU Programming Contest

problems by TaoSama and Heart_Blue, organized by TaoSama

2017 年 5 月 21 日

目录

Problem A. Arising Stars

Input file:	**standard input**
Output file:	**standard output**
Time limit:	1 second
Memory limit:	256 megabytes

题目描述

嗨呀，情侣一起看星星好浪漫的哦，蓝儿重庆没有星星 (冷漠脸

有一天韬韬和妹子去看星星了 (当然是在梦里，韬韬哪里有妹子哦)，哇，在草坪上铺一波毯子，又双叒叕头靠着一起，打开 *Google Sky Map* 看星图哦，能看一晚不碎觉！

星星一闪一闪的，妹子侧过头来指着一片星星问韬韬，这些星星是不是越来越亮的呀。韬韬瞄了一眼发现，并不是呢。韬韬害怕妹子不开心，就找了一找，并没有一条连成一条线的星星是越来越亮的。

韬韬陷入了沉思，妹子突然指着一片星星对韬韬说: “哇，它们是连成一条线的呀。那，它们有多少颗是相比前一颗更亮的呢？” “这个问题好简单啊，” 韬韬想道，“如果可以任意修改一颗星星的亮度，并且只修改一次，那么最多又有多少颗星星是比前一颗更亮的呢？”

其中**有多少颗是相比前一颗更亮**的一般表示为：给定 n 颗在一条线上的星星亮度数组 a，问有多少个 i 满足 $a_i > a_{i-1},\ i \in [2,\ n]$。

“看星图就看星图嘛，为啥要思考这样的问题呢？赶紧换个话题”，韬韬想道。“哇，快看，天猫座！”，韬韬突然说道。韬韬和妹子浪漫地去看星图了，亲爱的小伙伴你们知道这两个问题的答案嘛？

输入格式

输入包含多组数据。第一行为一个整数 $T(1 \le T \le 10)$，代表数据组数，对于每组数据：

第一行为一个整数 $n(2 \le n \le 100)$，代表一条线上星星的颗数

第二行为 n 个空格间隔的整数 $a_i(1 \le a_i \le 100)$，保证 a_i 各不相同，当然修改之后也必须保证 $1 \le a_i \le 100$。

输出格式

每组数据输出一行，为 2 个整数，其中第 1 个为妹子问题的答案，第 2 个为韬韬问题的答案。

测试样例

standard input	standard output
2 5 1 2 3 4 5 5 1 5 3 4 6	4 4 3 4

Problem B. Breadth First Search of Tree Vertices

Input file:	**standard input**
Output file:	**standard output**
Time limit:	3 seconds
Memory limit:	256 megabytes

题目描述

美丽的大学城有 n 个学校，分别标号为 $1, 2, 3, \cdots, n$，它们之间由 $n-1$ 条无向道路相连，每条道路的长度为单位长度，其中第 i 条道路连接第 a_i 号和第 b_i 学校，保证学校之间互相连通，韬韬现在在 1 号学校，韬韬想知道以自己所在的学校为根形成的这棵树的一些问题。
其中有 q 个问题，每个问题为：以 u 为根的子树里，距离韬韬为 d 的学校有几个。韬韬在大学城呆了两年呢，当然知道问题的答案啦，亲爱的小伙伴你知道嘛？

输入格式

输入包含多组数据。第一行为一个整数 $T(1 \le T \le 100)$，代表数据组数，对于每组数据：
第一行为一个整数 $n(1 \le n \le 10^5)$，接下来 n 行，每行两个空格分隔的整数 $a_i,\ b_i(1 \le a_i,\ b_i \le n)$
之后一行为一个整数 $q(1 \le q \le 10^5)$，接下来 q 行，每行两个空格分隔的整数 $u,\ d$，输入的 $u,\ d$ 是加密的，需经操作 $u = u \oplus lastans,\ d = d \oplus lastans$ 解密，解密后的 $u,\ d(1 \le u,\ d \le n)$（其中 $\oplus$ 表示按位异或运算，$lastans$ 表示上个问题的答案，初始 $lastans = 0$）。
保证 $n,\ q > 10^3$ 的数据不超过 10 组

输出格式

每组数据输出 q 行，每行一个整数，为问题的答案

测试样例

standard input	standard output
1 3 1 2 1 3 3 1 1 0 3 3 3	2 1 0

样例解释

样例一的解密后的询问为：
1 1
2 1
2 2

Problem C. Combo of Iori Yagami

Input file:	**standard input**
Output file:	**standard output**
Time limit:	1 second
Memory limit:	256 megabytes

题目描述

The King of Fighters！没有人能在韬韬的一套 Combo 中活下来，如果有，那么就两套。韬韬最强辣！
每个少年都喜欢 Iori 呀，韬韬选人的时候从来不会把他放到第一出场，不然可怜的小伙伴们是见不到韬韬的第二个人物的。
嘛，作为一个 Combo Master，韬韬熟悉 Iori 的各种连招！7 连八稚女！9 连八稚女！10 连八稚女！11 连八稚女！嗨呀，11 连八稚女好难的呢！
当然这些不重要，首先一套 Combo 由 >1 个 Hits 组成，假设我们 Iori 是左手站位，9 个方向由数字 $1-9$ 表示，4 种攻击方式由字母 $ABCD$ 表示，分别为轻拳，重拳，轻脚，重脚。为了简化题目，除特殊连招外，一个 Hit 为 1 个或者多个方向紧跟 1 个攻击方式组成，没有单个 $ABCD$ 的情况，并且单个方向不算 Hit，比如 $2C$ 和 $24C$ 都为 1 Hit，假设特殊连招 (简化连招) **只有**以下这些：

- $84C$ 是百合斩，1 Hit
- $626A$ 是鬼烧，3 $Hits$
- $6AA$ 是梦弹二式，2 $Hits$
- $624A$ 是葵花，1 Hit，但是葵花有三段，$(624A)\times 2$ 是葵花二式，2 $Hits$，$(624A)\times 3$ 是葵花三式，3 $Hits$，其中 $\times$ 用小写字母 x 表示
- $2426A$ 是八酒杯，1 Hit
- $2624A$ 是八稚女，8 Hit
- $2626AC$ 是豺华，7 $Hits$

韬韬当然知道自己一套 Combo 打了多少 $Hits$ 呢，那么给定韬韬的一套出招表亲爱的小伙伴你知道韬韬一共打了多少 $Hits$ 嘛？

输入格式

输入包含多组数据。第一行为一个整数 $T(1\le T\le 10)$，代表数据组数，对于每组数据：
第一行为一个字符串 $S(|S|\le 100)$，除 Combo 中有葵花二式和三式时含有括号以外，保证字符串仅由数字 $1-9$ 和字母 $A-D$ 组成，并且保证数据合法

输出格式

每组数据输出一行，为韬韬一套 Combo 打的 $Hits$ 数，如果不是一套 Combo，那么输出“Oops”(不包含引号)。

测试样例

standard input	standard output
2 6C6AA(624A)x3 84C3C56A6AA(624A)x22624A2626AC	6 22

样例解释

其中第二个样例为，百合斩接下后轻脚，回正，一个轻拳接梦弹二式，再接葵花二式，卡一个奇妙的时间接八稚女和豺华，一共 $1+1+1+2+2+8+7=22$ $Hits$，这就是其中一种 7 连八稚女咯

Problem D. Dark Souls III

Input file:	**standard input**
Output file:	**standard output**
Time limit:	1 second
Memory limit:	256 megabytes

Description

In Chinese, it is said that it doesn’ t affect the reading experience even if the literal is in arbitrary order. So what if in English? Anyway, it will be not as hard as you think, because it is just that the letters of each word is **in reversing order**.

amaSoaT si dessesbo ni a emag deman **kraD sluoS III** yltnecer. tI si ooooooos lufniap gniyalp siht emag, yreve emit amaSoaT sdeen ot etaluclac fi eh nac tih eht ymene ni eno ekirts. ecnO amaSoaT si deyals yb eht ymene, eh sah ot tratser eht emag, yllausu, niaga dna niaga.

oS amaSoaT si gnimoceb erom dna erom ylluferac ni gnitaluclac... dna amaSoaT si detsuahxe oot.

teL s’amaSoaT retcarahc dna ymene eb owt selgnatcer, nehw amaSoaT syrt ot tih eht ymene, a wen elgnatcer smrof. gnisoppuS taht eht ymene si llits, won yb gnivig uoy s’amaSoaT wen gnittih elgnatcer dna s’ymene elgnatcer, nac uoy pleh amaSoaT egduj fi ti si a tih ro a ssim?

A tih si taht s’amaSoaT gnittih elgnatcer sah noitcesretni htiw s’ymene elgnatcer, **neve fi tsuj yb egde ro xetrev**, esiwrehto ti si a ssim.

Input

tupnI sniatnoc elpitlum tset sesac. ehT tsrif enil si eht rebmun fo tset sesac $T(1 \le T \le 10^5)$, rof hcae fo tset sesac:

tI ylno sniatnoc eno enil, ereht era 8 sregetni, $x_1,\ y_1,\ x_2,\ y_2,\ x_3,\ y_3,\ x_4,\ y_4$ $(|x_i|,\ |y_i| \le 10^9,\ i \in [1,4])$ $(x_1,\ y_1)$, $(x_2,\ y_2)$ stneserper eht owt secitrev fo eht lanogaid fo s’amaSoaT gnittih elgnatcer, $(x_3,\ y_3)$, $(x_4,\ y_4)$ stneserper eht owt secitrev fo eht lanogaid fo s’ymene elgnatcer. tI si deetnaraug taht eht aera fo hcae elgnatcer si evitisop.

Output

tuptuO eno enil rof hcae fo tset sesac. fI amaSoaT stih eht ymene yllufsseccus, tuptuo ”tiH”(tuohtiw etouq), esiwrehto tuptuo ”ssiM”(tuohtiw etouq).

Sample

standard input	standard output
2 0 0 1 1 1 0 2 1 0 0 1 1 2 0 3 1	Hit Miss

Problem E. Easy Expression Parser

Input file:	**standard input**
Output file:	**standard output**
Time limit:	1 second
Memory limit:	256 megabytes

题目描述

据说大三的孩纸们有好多硬件课，天天玩板子，韬韬有一天得到了一个神奇的板子，它的 LED 显示屏是 $5 \times \infty$ 的。韬韬最喜欢表达式了，所以他打算烧一个计算器上去。

由于是 LED 显示屏，数字和符号都是由“ ”(空格) 和“*”组成的 $5 \times x$ 的点阵形式，下面给出它们具体的点阵形式表示：

数字 0 点阵

```
***
* *
* *
* *
***
```

数字 1 点阵

```
 *
 *
 *
 *
 *
```

数字 2 点阵

```
***
  *
***
*
***
```

数字 3 点阵

```
***
  *
***
  *
***
```

数字 4 点阵

```
* *
* *
***
  *
  *
```

数字 5 点阵

```
***
*
***
  *
***
```

数字 6 点阵

```
***
*
***
* *
***
```

数字 7 点阵

```
***
  *
  *
  *
  *
```

数字 8 点阵

```
***
* *
***
* *
***
```

数字 9 点阵

```
***
* *
***
  *
***
```

符号 + 点阵

```
   
 *
***
 *
   
```

符号 - 点阵

```
   
   
***
   
   
```

符号 * 点阵

```
   
* *
 * 
* *
   
```

符号 / 点阵

```
   
  *
 * 
*  
   
```

符号 (点阵

```
 *
*  
*  
*  
 *
```

符号) 点阵

```
 *
  *
  *
  *
 *
```

符号 = 点阵

```
     
*****
     
*****
     
```

韬韬就打算让这个计算器支持普通的运算，就是含有括号和加减乘除的中缀表达式啦，其中中缀表达式定义为运算符在运算数中间的表达式，即我们日常生活中用的算术表达式。加减乘除括号的符号表示分别为“+-*/()”，比如1+2/(3*4)=7/6
韬韬最近忙着做毕设，所以他打算找亲爱的小伙伴你来帮忙，使得这个板子的 LED 显示屏能完美的显示韬韬想要的效果，韬韬想要的效果如下：

- 最终 LED 屏显示的结果为一个等式，其中“=”号左侧为原来的表达式，右侧为表达式运算结果
- 表达式运算结果为一个既约分数 p/q，即 $gcd(p,\ q) = 1$，且当 $q = 1$ 时，仅输出 p
- 其中显示的等式中的所有的数字和符号为上图表格所示的点阵形式
- 并且每两个数字和符号之间间隔 5×2 的空格
- 具体可以参考测试样例

输入格式

输入包含多组数据。第一行为一个整数 $T(1 \le T \le 10)$，代表数据组数，对于每组数据：
包括一行，为一个字符串表达式 $S(|S| \le 10^5)$，保证表达式合法，其中数字不含前导 0，运算过程不会除以 0，且运算过程和运算结果在有符号整型数范围内，即 C++ 或 Java 语言的 int 类型
$\sum |S| \le 10^5$

输出格式

每组数据输出一行，为韬韬想要的 LED 屏上的显示结果。

测试样例

standard input	standard output
1 1+2/(3*4)	* *** * *** * * * *** *** * * * * * * * * * * * ****** * * * * *** *** * * *** * *** * * * *** * * * * * * * * * * ****** * * * * * *** * *** * * *** *

Problem F. Frustrating Matrix

Input file:	**standard input**
Output file:	**standard output**
Time limit:	5 seconds
Memory limit:	256 megabytes

题目描述

“矩阵好难啊。”，韬韬看着线代书想道。韬韬突然看到了一个有趣的题目，有一个 $n \times m$ 的矩阵 A，还有一个 $1 \times n$ 向量 B，矩阵和向量的下标均为 $0-based$ 的，定义 $C[i] = \sum_{j=0}^{n-1} A[j][B[i \times j \ mod \ n]]$，求 $C[i],\ i \in [0,n)$ 是多少。
韬韬想了很久很久，还是不会，亲爱的小伙伴们你们可以帮帮韬韬嘛？

输入格式

输入包含单组数据：
第一行为两个空格分隔的整数 $n(1 \le n \le 3 \times 10^5),\ m(1 \le m \le 4)$
接下来 n 行，每行有 m 个空格分隔的整数 $A[i][j](0 \le A[i][j] \le 2000)$
接下来一行为 n 个空格分隔的数字 $B[i](0 \le B[i] \le m)$

输出格式

每组数据输出一行，为 n 个空格分隔的整数 $C[i]$

测试样例

standard input	standard output
5 3 3 6 1 2 5 3 7 5 1 6 3 0 8 6 7 0 1 2 1 0	26 20 22 26 16

Problem G. Grouping the Toys

Input file:	**standard input**
Output file:	**standard output**
Time limit:	1 second
Memory limit:	256 megabytes

题目描述

重庆又迎来了一年一度的梅雨季节，雷声隆隆，雨点滚滚。前天韬韬拿拼图出去玩的时候，明明几分钟前是个晴天，突然来了一波暴雨，拼图全湿了。不过今天是个晴天，韬韬出去晒拼图了。

蓝儿韬韬的运气最近总是不好，突然又刮了大风，眼见着拼图块都要被吹走了，韬韬如果把它们都收回去的话太麻烦了呢。因为风并不会刮太久，之后放晴了还得拿出来晒。怎么办呢？韬韬灵机一动想到了一个好办法。

韬韬一共有 n 个拼图块，从左到右分别标号为 $1, 2, 3, \cdots, n$，每个拼图块有个坚固值 a_i，如果把连续的一段拼图块 $a_l, a_{l+1}, a_{l+2}, \cdots, a_r$ 拼成一个拼图大块，那么这个拼图大块的坚固值的为 $\sum_{i=l}^{r} a_i$，每个拼图块只能被拼一次，假设拼成了 m 个拼图大块，且它们的坚固值为 $b_1, b_2, b_3, \cdots, b_m$。因为风是从左往右吹的啦，如果一个拼图大块的坚固值 $\leq$ 它右边紧跟着的拼图大块的坚固值，那么它就不会被吹走啦。也就说如果满足 $b_i \leq b_{i+1},\ i \in [1,\ m)$，那么这些拼图就是坚固哒，不容易被风吹走。并且 m 越大越坚固越不容易被风吹走。

韬韬当然知道怎样把拼图块拼成拼图大块，并且使得它们是坚固的 m 的最大值啦，亲爱的小伙伴你知道吗？

输入格式

输入包含多组数据。第一行为一个整数 $T(1 \leq T \leq 100)$，代表数据组数，对于每组数据：

第一行是一个整数 $n(1 \leq n \leq 300)$，接下来一行为 n 个空格分隔的整数 $a_i(1 \leq a_i \leq 10^5)$

保证 $n > 100$ 的数据不超过 10 组

输出格式

每组数据输出一行，为 m 的最大值

测试样例

standard input	standard output
2 5 1 2 3 4 5 8 10 4 3 9 5 3 5 7	5 3

样例解释

其中第二个样例的一个最优解为 10, 4 3 9, 5 3 5 7，$10 \leq 16 \leq 20$，一共 3 组

Problem H. Harmonious Buildings of Campus Huxi

Input file:	**standard input**
Output file:	**standard output**
Time limit:	3 seconds
Memory limit:	256 megabytes

题目描述

美丽的虎溪校区有 n 个教学楼，分别标号为 $1,2,3,\cdots,n$，还有 m 个学院，分别标号为 $1,2,3,\cdots,m$，我们知道每个教学楼属于且仅属于一个学院。

n 个教学楼之间由 $n-1$ 条无向道路相连，每条道路的长度为单位长度，其中第 i 条道路连接第 a_i 号和第 b_i 号教学楼，保证教学楼之间互相连通。

学校最近搞起了什么非限制性选修，必须要选别的学院的课，所以就很烦，韬韬想知道每个学院里距离最远的 2 个教学楼之间的距离，这样就可以避开那些上了这节下一节换教学楼就需要走好远路的课呢。

韬韬当然知道这个问题的答案啦，因为韬韬很懒并不想走远路的。亲爱的小伙伴们你知道这个问题的答案嘛？

输入格式

输入包含多组数据。第一行为一个整数 $T(1 \le T \le 50)$，代表数据组数，对于每组数据：

第一行为 2 个空格间隔的整数 $n(2 \le n \le 10^5)$ 和 $m(1 \le m \le 10^4)$

之后一行为 n 个空格间隔的整数 $c_i(1 \le c_i \le m)$，表示第 i 号教学楼属于第 c_i 号学院。

接下来 $n-1$ 行，每行 2 个空格间隔的整数 a_i 和 b_i，$(1 \le a_i,\ b_i \le n)$。

保证每个学院至少含有 2 个教学楼，即一定有距离。保证 $n > 1000$ 的数据不超过 25 组

输出格式

每组数据输出一行，为 m 个空格分隔的整数。

测试样例

standard input	standard output
2 2 1 1 1 1 2 5 2 1 2 2 1 1 1 2 2 3 3 4 4 5	1 4 1

Problem I. Increasing Blocks of WordMao's File System

Input file:	**standard input**
Output file:	**standard output**
Time limit:	3 seconds
Memory limit:	256 megabytes

题目描述

韬韬有个好朋友天猫，天猫总说自己是个写项目特别渣的蒟蒻，于是他总是幻想自己能写出很腻害东西。有一天天猫又开始 YY，以下是天猫的 YY：天猫写了一个文件管理系统，名叫“Word 猫”。这个管理系统可以按文件名的字典序放置文件，比如现在有 n 个已经排好顺序的文件，第 i 个文件名有 a_i 个字符，这个管理系统的界面有若干块的，每块都有 x 行，每行放 1 个文件，因此每块的宽度等于当前块中最长的文件名长度，记为 a_{maxi}。天猫将文件从前至后依次排满每一块，当然最后一块不一定能放满。由于每一块是并列显示在桌面上的，两块之间用一个空格字符隔开，所以这个文件管理系统显示出来的总宽度就是 $(\sum(a_{maxi}+1))-1$

现在天猫想要知道，如果有 n 个已经排好序的文件，把他们的文件名都显示在屏幕上（即总宽度不超过屏幕宽度 w），每块最少需要多少行？韬韬最强辣！他当然知道答案了，可是他现在有点忙，亲爱的小伙伴们你可以帮帮天猫嘛？

输入格式

输入包含多组数据。第一行为一个整数 $T(1 \le T \le 100)$，代表数据组数，对于每组数据：

第一行为 2 个空格间隔的整数 $n(1 \le n \le 10^5)$ 和 $w(1 \le w \le 10^9)$

之后一行为 n 个空格分隔的整数 $a_i(1 \le a_i \le w)$

输出格式

每组数据输出一行，包含一个整数 x，为每块最少需要的行数。

测试样例

standard input	standard output
1 11 20 1 3 7 4 1 2 1 1 1 1 4	3

样例解释

其中第一个样例的最优解为，x 是 3 行，一共是 4 块，宽度分别为 $7,4,1,4$，那么总宽度为 $7+4+1+4+3=19 \le w=20$