

2005 年百度之星程序设计大赛试题初赛题目

第一题（共四题 100 分）：连续正整数（10 分）

题目描述：一个正整数有可能可以被表示为 $n(n \geq 2)$ 个连续正整数之和，如：

$$15=1+2+3+4+5$$

$$15=4+5+6$$

$$15=7+8$$

请编写程序，根据输入的任何正整数，找出符合这种要求的所有连续正整数序列。

输入数据：一个正整数，以命令行参数的形式提供给程序。

输出数据：在标准输出上打印出符合题目描述的全部正整数序列，每行一个序列，每个序列都从该序列的最小正整数开始、以从小到大的顺序打印。如果结果有多个序列，按各序列的最小正整数的大小从小到大打印各序列。此外，序列不允许重复，序列内的整数用一个空格分隔。如果没有符合要求的序列，输出“NONE”。

例如，对于 15，其输出结果是：

1 2 3 4 5

4 5 6

7 8

对于 16，其输出结果是：

NONE

评分标准：程序输出结果是否正确。

百度之星程序设计大赛试题 -2

第二题（共四题 100 分）：重叠区间大小（20 分）

题目描述：请编写程序，找出下面“输入数据及格式”中所描述的输入数据文件中最大重叠区间的大小。

对一个正整数 n ，如果 n 在数据文件中某行的两个正整数（假设为 A 和 B ）之间，即 $A \leq n \leq B$ 或 $A \geq n \geq B$ ，则 n 属于该行；如果 n 同时属于行 i 和 j ，则 i 和 j 有重叠区间；重叠区间的大小是同时属于行 i 和 j 的整数个数。

例如，行（10 20）和（12 25）的重叠区间为 [12 20]，其大小为 9；行（20 10）和（12 18）的重叠区间为 [10 12]，其大小为 3；行（20 10）和（20 30）的重叠区间大小为 1。

输入数据：程序读入已被命名为 input.txt 的输入数据文本文件，该文件的行数在 1 到 1,000,000 之间，每行有用一个空格分隔的 2 个正整数，这 2 个正整数的大小次序随机，每个数都在 1 和 $2^{32}-1$ 之间。（为便于调试，您可下载测试 input.txt 文件，实际运行时我们会使用不同内容的输入文件。）

输出数据：在标准输出上打印出输入数据文件中最大重叠区间的大小，如果所有行都没有重叠区间，则输出 0。

评分标准：程序输出结果必须正确，内存使用必须不超过 256MB，程序的执行时间越快越好。

百度之星程序设计大赛试题 -3

第三题（共四题 100 分）：字符串替换（30 分）

题目描述：请编写程序，根据指定的对应关系，把一个文本中的字符串替换成另外的字符串。

输入数据：程序读入已被命名为 `text.txt` 和 `dict.txt` 的两个输入数据文本文件，`text.txt` 为一个包含大量字符串（含中文）的文本，以 `whitespace` 为分隔符；`dict.txt` 为表示字符串（`s1`）与字符串（`s2`）的对应关系的另一个文本（含中文），大约在 1 万行左右，每行两个字符串（即 `s1` 和 `s2`），用一个 `\t` 或空格分隔。`dict.txt` 中各行的 `s1` 没有排序，并有可能有重复，这时以最后出现的那次 `s1` 所对应的 `s2` 为准。`text.txt` 和 `dict.txt` 中的每个字符串都可能包含除 `whitespace` 之外的任何字符。`text.txt` 中的字符串必须和 `dict.txt` 中的某 `s1` 完全匹配才能被替换。（为便于调试，您可下载测试 `text.txt` 和 `dict.txt` 文件，实际运行时我们会使用不同内容的输入文件。）

输出数据：在标准输出上打印 `text.txt` 被 `dict.txt` 替换后的整个文本。

评分标准：程序输出结果必须正确，内存使用越少越好，程序的执行时间越快越好。

第四题（共四题 100 分）：低频词过滤（40 分）

题目描述：请编写程序，从包含大量单词的文本中删除出现次数最少的单词。如果有多 个单词都出现最少的次数，则将这些单词都删除。

输入数据：程序读入已被命名为 `corpus.txt` 的一个大数据量的文本文件，该文件包含英文单词和中文单词，词与词之间以一个或多个 `whitespace` 分隔。（为便于调试，您可下载测试 `corpus.txt` 文件，实际运行时我们会使用不同内容的输入文件。）

输出数据：在标准输出上打印删除了 `corpus.txt` 中出现次数最少的单词之后的文本（词与词保持原来的顺序，仍以空格分隔）。

评分标准：程序输出结果必须正确，内存使用越少越好，程序的执行时间越快越好。

2005 年百度之星程序设计大赛试题总决赛题目

题目描述：

八方块移动游戏要求从一个含 8 个数字（用 1-8 表示）的方块以及一个空格方块（用 0 表示）的 3x3 矩阵的起始状态开始，不断移动该空格方块以使其和相邻的方块互换，直至达到所定义的目标状态。空格方块在中间位置时有上、下、左、右 4 个方向可移动，在四个角落上有 2 个方向可移动，在其他位置上有

3 个方向可移动。例如，假设一个 3x3 矩阵的初始状态为：

8 0 3

2 1 4

7 6 5

目标状态为：

1 2 3

8 0 4

7 6 5

则一个合法的移动路径为：

8 0 3 8 1 3 8 1 3 0 1 3 1 0 3 1 2 3

2 1 4 => 2 0 4 => 0 2 4 => 8 2 4 => 8 2 4 => 8 0 4

7 6 5 7 6 5 7 6 5 7 6 5 7 6 5 7 6 5

另外，在所有可能的从初始状态到目标状态的移动路径中，步数最少的路径被称为最短路径；在上面的例子中，最短路径为 5。如果不存在从初试状态到目标状态的任何路径，则称该组状态无解。请设计有效的（细节请见评分规则）算法找到从八方块的某初试状态到某目标状态的所有可能路径中的最短路径，并用 C/C++ 实现。

输入数据：

程序需读入已被命名为 start.txt 的初始状态和已被命名为 goal.txt 的目标状态，这两个文件都由 9 个数字组成（0 表示空格，1-8 表示 8 个数字方块），每行 3 个数字，数字之间用空格隔开。

输出数据：

如果输入数据有解，输出一个表示最短路径的非负的整数；如果输入数据无解，输出 -1。

自测用例：

如果输入为：start.txt 和 goal.txt，则产生的输出应为：

5

又例，如果用

7 8 4

3 5 6

1 0 2

替换 start.txt 中的内容，则产生的输出应为：

21

评分规则：

1) 我们将首先使用和自我用例不同的 10 个 start.txt 以及相同的 goal.txt ，每个测试用例的运行时间在一台 Intel Xeon 2.80GHz 4 CPU/ 6G 内存的 Linux 机器上应不超过 10 秒（内存使用不限制），否则该用例不得分；

2) 每个选手的总分（精确到小数点后 6 位）=10 秒钟内能产生正确结果的测试用例数量 $\times 10^{-4}$ （ $1/10^4$ 产生这些正确结果的测试用例的平均运行毫秒 ）；

3) 如果按此评分统计仍不能得出总决赛将决出的一、二、三等奖共计九名获奖者，我们将先设 $N=2$ ，然后重复下述过程直至产生最高的 9 位得分：用随机生成的另外 10 个有解的 start.txt 再做测试，并对这 $10 \times N$ 个测试用例用 2) 中公式重新计算总分， $N++$ 。

2006 年百度之星程序设计大赛初赛题目 1

饭团的烦恼

“午餐饭团”是百度内部参与人数最多的民间组织。

同一个部门的，同一间大学的，同一年出生的，用同一种型号电脑的，员工们总是以各种理由，各种借口组织各种长久的，临时的饭团。

参加饭团，不仅可以以优惠的价格尝到更加丰富的菜式，还可以在吃饭的时候和同事们唠唠嗑，吹吹水，增进感情。

但是，随着百度的员工越来越多，各个饭团的管理随即变得烦杂。特别是为了照顾员工们越来越挑剔的胃口，饭团的点菜负责人背负的责任越来越大。现在，这个重担落在百度之星的肩上，因为，你们将要为所有的百度饭团设计一个自动点菜的算法。

饭团点菜的需求如下：

1. 经济是我们要考虑的一个因素，既要充分利用百度员工的午餐补助，又不能铺张浪费。因此，我们希

望最后的人均费用越接近 12 元越好。

2. 菜式丰富是我们要考虑的另一个因素。为简单起见，我们将各种菜肴的属性归结为荤菜，素菜，辛辣，清淡，并且每个菜只能点一次。

3. 请紧记，百度饭团在各大餐馆享受 8 折优惠。

输入数据描述如下：

第一行包含三个整数 N, M, K ($0 \leq N \leq 16, 0 \leq M \leq N, 0 \leq K \leq 12$)，分别表示菜单上菜的数目，饭团需要点的菜的数目，就餐的人数。 $K \leq 12$)，分别表示菜单上菜的数目，饭团需要点的菜的数目，就餐的人数。

紧接着 N 行，每行的格式如下：

菜名（长度不超过 20 个字符） 价格（原价，整数） 是否荤菜（1 表示是，0 表示否） 是否辛辣（1 表示是，0 表示否） 例：

水煮鱼 30 1 1

紧接着是 a, b, c, d 四个整数，分别表示需要点的荤菜，素菜，辛辣，清淡菜的数目。

输出数据：

对于每一测试数据，输出数据包含 $M+1$ 行，前 M 行每行包含一个菜名（按菜名在原菜单的顺序排序）。第 $M+1$ 行是人均消费，结果保留两位小数。

说明：

1. 结果菜单的数目应该恰好为 M ，荤菜，素菜，辛辣，清淡菜的数目恰好为 a, b, c, d 。在满足这样的前提下，选择人均消费最接近 12 元的点菜方案。题目数据保证有且仅有一个解。

2. 每组测试数据的结果用一个空行隔开。末尾不要有多余的空行。

输入样例

3 2 2

水煮鱼 30 1

1 口水鸡 18 1 1

清炖豆腐 12 0 0

1 1 1 1

输出样例

口水鸡

清炖豆腐

12.00

时间要求：1S 之内

2006 年百度之星程序设计大赛初赛题目 2

题目名称：蝓蝓式的记分

内容描述：

蝓蝓小朋友刚刚学会了 0-9 这十个数字,也跟爸爸妈妈来参加百度每周进行的羽毛球活动。但是他还没有球拍高,于是大人们叫他记录分数。聪明的蝓蝓发现只要记录连续得分的情况就可以了,比如用“3 2 4”可以表示一方在这一局中连得三分后,输了两分,接着又连得到四分。可是,后来大人们发现蝓蝓只会用 0-9 这十个数字,所以当比赛选手得分超过 9 的时候,他会用一个 X 来表示 10 完成记分。但问题是,当记录为“X 3 5”的时候,蝓蝓自己也记不起来是一方连续得到十三分后,再输五分;还是先赢十分输三分再赢五分。

因为百度内部就要开始进行羽毛球联赛了,要先摸清大家的实力才好分组比赛呢~于是,大人们想知道以前每局的比分是怎样的,以及谁获得了胜利。要是遇到了根据比赛记录无法确认比赛进程的情况,也要输出相应的提示哦。

需要帮蝓蝓进一步说明的是,比赛是五局三胜的,每局先获得二十一分的为胜,但是胜方必须领先对手两分或以上,否则必须继续比赛直到一方超出对手两分为止,比分多的一方获胜。任何一方先获得三局的胜利后就获得胜利,比赛也相应的结束。而且蝓蝓保证是完整的无多余信息的记录了比赛。

输入数据:

以 point.in 为输入文件,文件中首行只有一个整数 M,表示蝓蝓记录了多少场比赛的分数。每场比赛用两行记录,第一行是一个整数 N(N<=1000)表示当前这个记录中有多少个字符,第二行就是具体的 N 个字符表示记录的分数。

输出数据:

相应的内容将输出到 point.out 文件中,对应每一个分数记录,输出相应的每局分数,每局分数都使用两个整数表示,表示两个选手的得分,中间用“:”分隔开;每组分分数记录间使用一个空行分隔开。如果相应的比赛结果无法预测的时候,以“Unknown”一个单词独占一行表示。

输入和输出结果数据样例:

Sample Input:

3

23

```

9 7 3 6 2 4 7 8 3 2 7 9 X 2 2 1 2 1 X 1 X 1 1
25
9 3 8 5 4 8 3 9 8 4 X X X X 2 X X X X 2 8 4 9 2 4
43
7 7 7 7 7 3 4 5 6 7 6 5 4 2 1 3 5 7 9 7 5 3 1 3 0 9 9 3 9 3 2 1 1 1 5 1 5 1 5 1 5 5 1
Sample Output:
21:17
24:22
21:3
Unknown
21:14
20:22
21:23
21:16
21:9

```

2006 年百度之星程序设计大赛初赛题目 3

变态的比赛规则

为了促进各部门员工的交流，百度(baidu)举办了一场全公司范围内的“拳皇友谊赛”，负责组织这场比赛的是百度的超级“拳皇”迷 W. Z. W. Z 不想用传统的淘汰赛或者循环赛的方式，而是自己制定了一个比赛规则。

由于一些员工（比如同部门或者相邻部门员工）平时接触的机会比较多，为了促进不同部门之间的交流，W. Z 希望员工自己组成不同组。不同组之间的每两个人都会进行一场友谊赛而同一组内的人则之间不会打任何比赛。

比如 4 个人，编号为 1—4, 如果分为两个组并且 1, 2 一个组，3, 4 一个组，那么一共需要打四场比赛：1 vs 3, 1 vs 4, 2 vs 3, 2 vs 4. 而如果是 1, 2, 3 一组，4 单独一组，那么一共需要打三场比赛：1 vs 4, 2 vs 4, 3 vs 4.

很快 W. Z 意识到，这样的比赛规则可能会让比赛的场数非常多。W. Z 想知道如果有 N 个人, 通过上面这种比赛规则，总比赛场数有可能为 K 场吗？比如 3 个人，如果只分到一组则不需要比赛，如果分到两组则需要 2 场比赛, 如果分为三组则需要 3 场比赛。但是无论怎么分都不可能只需要 1 场比赛。

相信作为编程高手的你一定知道该怎么回答这个问题了吧？ 那么现在请你帮助 W. Z 吧。

输入

每行为一组数据，包含两个数字 N, K。 ($0 \leq N \leq 500$, $K \geq 0$)

输出

对输入的 N, K 如果 N 个员工通过一定的分组方式可能会一共需要 K 场比赛，则输出“YES”，否则输出“NO”，每组数据占一行。所有的输入输出均为标准输入输出。

例子

输入文件：

2 0

2 1

3 1

3 2

输出：

YES

YES

NO

YES

2006 年百度之星程序设计大赛初赛题目 4

2007-05-14 17:39

剪刀石头布

N 个小孩正在和你玩一种剪刀石头布游戏。N 个小孩中有一个是裁判，其余小孩分成三组（不排除某些组没有任何成员的可能性），但是你不知道谁是裁判，也不知道小孩们的分组情况。然后，小孩们开始玩剪刀石头布游戏，一共玩 M 次，每次任意选择两个小孩进行一轮，你会被告知结果，即两个小孩的胜负情况，然而你不会得知小孩具体出的是剪刀、石头还是布。已知各组的小孩分别只会出一种手势（因而同一组的两个小孩总会是和局），而裁判则每次都会随便选择出一种手势，因此没有人会知道裁判到底会出什么。请你在 M 次剪刀石头布游戏结束后，猜猜谁是裁判。如果你能猜出谁是裁判，请说明最早在第几次游戏结束后你能够确定谁是裁判。

输入格式：

输入文件包含多组测试数据。每组测试数据第一行为两个整数 N 和 M ($1 \leq N \leq 500$, $0 \leq M \leq 2000$)，分别为小孩的个数和剪刀石头布游戏进行的次数。接下来 M 行，每行两个整数且中间以一个符号隔开。两个整数分别为进行游戏的两个小孩各自的编号，为小于 N 的非负整数。符号的可能值为“=”、“>”和“<”，分别表示和局、第一个小孩胜和第二个小孩胜三种情况。

输出格式：

每组测试数据输出一行，若能猜出谁是裁判，则输出身为裁判的小孩的编号，并输出在第几次游戏结束后就能够确定谁是裁判。如果无法确定谁是裁判，或者发现剪刀石头布游戏的胜负情况不合理（即无论谁是裁判都会出现矛盾），则输出相应的信息。具体输出格式请参考输出样例。

输入样例：

3 3

0<1

1<2

2<0

3 5

0<1
0>1
1<2
1>2
0<2
4 4
0<1
0>1
2<3
2>3
1 0

输出样例：

Can not determine

Player 1 can be determined to be the judge after 4 lines

Impossible

Player 0 can be determined to be the judge after 0 lines

说明：

共有 5 个测试数据集，每个测试数据集为一个输入文件，包含多组测试数据。每个测试数据集从易到难分别为 5、10、15、30 和 40 分，对每个测试数据集分别执行一次程序，每次必须在运行时限 3 秒内结束程序并输出正确的答案才能得分。

所有数据均从标准输入设备（stdin/cin）读入，并写出到标准输出设备（stdout/cout）中。

五个测试数据集中输入 N 分别不大于 20、50、100、200 和 500，各有 10 组测试数据。

2006 年百度之星程序设计大赛初赛题目 5

座位调整

题目描述：

百度办公区里到处摆放着各种各样的零食。百度人力资源部的调研发现，员工如果可以在自己喜欢的美食旁边工作，工作效率会大大提高。因此，百度决定进行一次员工座位的大调整。

调整的方法如下：

1. 首先将办公区按照各种零食的摆放分成 N 个不同的区域。（例如：可乐区，饼干区，牛奶区等等）。
2. 每个员工对不同的零食区域有不同的喜好程度（喜好程度度的范围为 1 — 100 的整数， 喜好程度越大表示该员工越希望被调整到相应的零食区域）。
3. 由于每个零食区域可以容纳的员工数量有限，人力资源部希望找到一个最优的调整方案令到总的喜好程度最大。

数据输入：

第一行包含两个整数 N, M, ($1 \leq N, M \leq 300$)。分别表示 N 个区域和 M 个员工。

第二行是 N 个整数构成的数列 a, 其中 $a[i]$ 表示第 i 个区域可以容纳的员工数, ($1 \leq a[i] \leq M$,

$a[1]+a[2]+\dots+a[N]=M$) 。

紧接着是一个 $M \times N$ 的矩阵 P , $P(i, j)$ 表示第 i 个员工对第 j 个区域的喜好度。

答案输出:

对于每个测试数据, 输出可以达到的最大的喜好程度。

输入样例:

```
3 3
1 1 1
100 50 25
100 50 25
100 50 25
```

输出样例:

175

数据解释: 此数据只存在一种安排方法, 三个员工分别安置在三个区域。最终的喜好程度为 $100+50+25=175$

2006 年百度之星程序设计大赛初赛题目 6

2007-05-14 17:42

百度语言翻译机

时限 1s

百度的工程师们是非常注重效率的, 在长期的开发与测试过程中, 他们逐渐创造了一套他们独特的缩率语。他们在平时的交谈, 会议, 甚至在各技术文档中都会大量运用。

为了让新员工可以更快地适应百度的文化, 更好地阅读公司的技术文档, 人力资源部决定开发一套专用的翻译系统, 把相关文档中的缩率语和专有名词翻译成日常语言。

输入数据:

输入数据包含三部分

1. 第一行包含一个整数 N ($N \leq 10000$), 表示总共有多少个缩率语的词条。
2. 紧接着有 N 行的输入, 每行包含两个字符串, 以空格隔开。第一个字符串为缩率语 (仅包含大写英文字符, 长度不超过 10), 第二个字符串为日常语言 (不包含空格, 长度不超过 255)。
3. 从第 $N+2$ 开始到输入结束为包含缩略语的相关文档。(总长度不超过 1000000 个字符)

输出数据:

输出将缩率语转换成日常语言的文档。(将缩率语转换成日常语言, 其他字符保留原样)

输入例子:

6

PS 门户搜索部

NLP 自然语言处理

PM 产品市场部

HR 人力资源部

PMD 产品推广部

MD 市场发展部

百度的部门包括 PS, PM, HR, PMD, MD 等等, 其中 PS 还包括 NLP 小组。

输出例子:

百度的部门包括门户搜索部, 产品市场部, 人力资源部, 产品推广部, 市场发展部等等, 其中门户搜索部还包括自然语言处理小组。

注意:

1. 输入数据中是中英文混合的, 中文采用 GBK 编码。
2. 为保证答案的唯一性, 缩率语的转换采用正向最大匹配(从左到右为正方向)的原则。请注意输入例子中 PMD 的翻译。

2006 年百度之星程序设计大赛复赛题目 1



另类杀人游戏

周末的晚上, 百度的员工们总喜欢聚集在公司的会议室玩杀人游戏。从 1 警 1 匪到 n 警 n 匪, 他们尝试了几乎所有流行的杀人游戏规则。终于有一天, 连最热衷杀人游戏, “杀人”不眨眼的 Austin 也开始对无休止的辩论感到厌烦。于是, 他决定改变他的一贯作风, 他开始变成了一个“杀人不睁眼”的杀手。

如何做到杀人不睁眼呢? Austin 早已构思好他的杀人计划:

1. N 个人(包括 Austin)坐成一圈玩杀人游戏, 按顺时针编号 1, 2, 3, 4, ...。
2. Austin 从 1 号开始顺时针开始数到第 m 号就杀掉第一个人。被杀掉的人要退出游戏。
3. 如果第 m 个人恰好是 Austin 自己, 他就杀掉他顺时针方向的下一个人。
4. Austin 从被杀的人的下一个顺时针数 m 个人, 把第 m 个杀掉。
5. 重复 2-4, 直至杀掉所有人。

Austin 把这个杀人计谋告诉了法官小 k, 他便可以闭起眼睛杀人啦。作为一个正直善良的法官, 小 k 当然不能让残忍的 Austin 得逞, 于是, 她偷偷把 Austin 的杀人计划告诉了作为警察的你, 聪明的百度之星。现在, 你的任务是活到最后, 与 Austin 单挑。

输入:

第一个行包含一个整数 T, 表示有 T 组测试数据。

对于每组测试数据:

三个整数

N, M, T , ($3 \leq N \leq 10000, 1 \leq M, T \leq 10000$) 分别表示参与游戏的人数, Austin 每隔 M 个人会杀掉一人, Austin 初始位置的标号。

输出:

每个测数数据输出一个整数。

你需要选择的初始位置的序号, 以确保最后剩下的两个人是你与 Austin。

输入例子:

2

7 4 1

7 4 1

输出例子

5

5

例子说明: 杀人顺序为 4 2 7 6 3 5, 所以 5 是你要选择的位置。

2006 年百度之星程序设计大赛复赛题目 2



空中飞猴

马戏团里新来了一只很特别的小猴子皮皮——不仅长得漂亮, 还很聪明。自从它来到马戏团之后, “空中飞猴”成了马戏团里保留节目, 慕名观看的人络绎不绝。“空中飞猴”表演开始时, 空中架着两根长长的钢丝。皮皮在其中一根上, 它的目标是到达另一个根钢丝上。皮皮必须在爬行一定距离后纵身一跃, 直接跳到另一根钢丝的某个位置。由于皮皮的速度非常快, 它的运动轨迹可以近似的看成一条直线段。为了不让自己太危险, 皮皮希望自己的跳跃距离尽量短, 而为了不让观众等得太不耐烦, 它在钢丝上的爬行距离不能超过 d 。在爬行距离不超过 d 的情况下, 皮皮的跳跃距离最短是多少?

输入格式:

输入文件包含多组测试数据。每组测试数据包含 16 个实数 $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4, x_p, y_p, z_p, d$, 表示两根钢丝分别为线段 $(x_1, y_1, z_1)-(x_2, y_2, z_2)$ 和 $(x_3, y_3, z_3)-(x_4, y_4, z_4)$, 皮皮的坐标为 (x_p, y_p, z_p) , 最大爬行距离为 d 。皮皮保证在第一条钢丝上, 保证每条钢丝长度大于零。但两条钢丝有可能相交甚至重叠。

输出格式:

每组测试数据输出一行, 仅包含一个非负实数, 四舍五入保留三位小数, 即最短跳跃距离。

输入样例：

0.0 0.0 0.0 4.0 4.0 0.0 4.0 0.0 1.0 0.0 4.0 1.0 2.0 2.0 0.0 10.0

输出样例：

1.000

说明：

共有 3 个测试数据集，每个测试数据集为一个输入文件，包含多组测试数据。每个测试数据集从易到难分别为 30、30 和 40 分，对每个测试数据集分别执行一次程序，每次必须在运行时限 3 秒内结束程序并输出正确的答案才能得分。

所有数据均从标准输入设备（stdin/cin）读入，并写出到标准输出设备（stdout/cout）中。

三个测试数据集各有 10000 组测试数据。

2006 年百度之星程序设计大赛复赛题目 3

星球大战

公元 4999 年，人类科学高度发达，绝大部分人都已经移居至浩瀚的宇宙，在上千颗可居住星球上留下了人类的印记。然而，此时人类却分裂成了两个联盟：正义联盟和邪恶联盟。两个联盟之间仇恨难解，时有战争。现在，正义联盟计划要破坏邪恶联盟的贸易网络，从而影响邪恶联盟的经济状况，为下一次战争作好准备。邪恶联盟由数百颗星球组成，贸易通过星球间的运输航道来完成。一条运输航道是双向的且仅连接两个星球，但两个星球之间可以有多条航道，也可能没有。两个星球之间只要有运输航道直接或间接的相连，它们就可以进行贸易。正义联盟计划破坏邪恶联盟中的一些运输航道，使得邪恶联盟的星球分成两部分，任一部分的星球都不能与另一部分的星球进行贸易。但是为了节省破坏行动所需的开支，正义联盟希望破坏尽量少的运输航道来达成目标。请问正义联盟最少需要破坏多少条运输航道呢？

输入格式：

输入文件包含多组测试数据。每组测试数据第一行为两个整数 N 和 M ($2 \leq N \leq 500$, $0 \leq M \leq N(N-1)/2$)， N 为邪恶联盟中星球的数量。接下来 M 行，每行三个整数 A 、 B 和 C ($0 \leq A, B < N$)，表示星球 A 和星球 B 之间有 C 条运输航道。运输航道的总数量不超过 108。

输出格式：

每组测试数据输出一行，包含一个整数，表示需要破坏的运输航道的数量。

如果输入的贸易网络本来就是不连通的，则输出 0。

输入样例：

3 3
0 1 1
1 2 1
2 0 1
4 3
0 1 1
1 2 1
2 3 1
8 14
0 1 1

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

输出样例：

2
1
2

说明：

共有 5 个测试数据集，每个测试数据集为一个输入文件，包含多组测试数据。每个测试数据集从易到难分别为 5、10、15、30 和 40 分，对每个测试数据集分别执行一次程序，每次必须在运行时限 10 秒内结束程序并输出正确的答案才能得分。

所有数据均从标准输入设备（stdin/cin）读入，并写出到标准输出设备（stdout/cout）中。

五个测试数据集中输入 N 分别不大于 20、50、100、200 和 500，各有 9 组测试数据。



2006 年百度之星程序设计大赛复赛题目 4

彩球游戏

X 博士是一个研究儿童智力开发方法的科学家，他为幼儿教育领域做出了许多贡献。最近，X 博士正在研究一种适合儿童的游戏，用以辅助发展儿童的观察力、注意力和思维能力。经过连日的构思，X 博士终于设计出了一种游戏：彩球游戏。

彩球游戏是一种单人参与的游戏，游戏首先给出一串由许多不同颜色的小球组成的小球序列，以及一个整数参数 M ($M \geq 2$)。一段连续的具有相同颜色的小球序列称为连续同色序列。小孩，即游戏参与者，每次可以向任意一段连续同色序列插入一个同色小球，使该序列的长度加一。当一段连续同色序列在插入一个同色小球后其长度达到 M 时，该序列就会爆炸消失，然后原序列两边的其余小球会重新连成一串，如果两

段相同颜色的连续同色序列在此时连接在一起，它们就会合并形成一段新的连续同色序列。如果新形成的连续同色序列长度达到 M ，这段序列也会爆炸消失，然后重复上述过程，直到没有新的长度达到 M 的连续同色序列出现为止。游戏的目标很简单，就是插入尽量少的小球，使得所有小球都爆炸消失掉。

通过长时间的游戏和不断提高游戏水平，这个游戏可以很好地开发儿童的观察力、注意力和思维能力。但是 X 博士仍然面临着一个困难的问题，他还需要设计出一个游戏演示 AI 程序，可以以最优的方式（即插入的小球数量最小）进行游戏，用于游戏教学，或者在游戏中对小孩给出提示。X 博士并不擅长此类程序，因而他无法完成这个任务，你可以帮助他吗？

输入格式：

输入文件包含多组测试数据。每组测试数据第一行为整数 M ($2 \leq M \leq 20$)，第二行为一条非空的字符串，由大写字母组成且长度不超过 200，表示初始的一串小球，不同的字母表示不同的小球颜色。初始时可能会存在一些长度达到 M 的连续同色序列，但这些序列不会马上爆炸消失。

输出格式：

每组测试数据输出一行，表示至少需要插入多少次小球才能使所有小球爆炸消失掉。

输入样例：

```
3
AAABAAA
3
ABBABBA
```

输出样例：

```
2
2
```

说明：

共有 5 个测试数据集，每个测试数据集为一个输入文件，包含多组测试数据。每个测试数据集从易到难分别为 5、10、15、30 和 40 分，对每个测试数据集分别执行一次程序，每次必须在运行时限 30 秒内结束程序并输出正确的答案才能得分。

所有数据均从标准输入设备（stdin/cin）读入，并写出到标准输出设备（stdout/cout）中。

五个测试数据集中输入初始小球队列的长度分别不大于 10、20、50、100 和 200，各有不超过 5000 组测试数据。



2006 年百度之星程序设计大赛复赛题目 5

追捕

四个小孩正在花园里玩追捕游戏。一个小孩扮演逃亡者，其余三个小孩做追捕者。花园是一块由 N 行 M 列方格组成的草地，花园周围有木栏包围着，不能走出，花园里面还有一些障碍物不能够通过。游戏可以进行许多回合，每个回合分成两轮，第一轮追捕者可以进行追捕行动，第二轮逃亡者可以根据前一轮追捕者的行动开展逃亡旅程。在第一轮里，三个追捕者必须在三人中选择一个人向某个相邻的方格走一步，只有在三个人都没有可以走的相邻方格时，他们才允许选择停留在原地。在第二轮里，逃亡者也必须选择某个

相邻的方格走一步，如果逃亡者没有任何可走的方格，那么逃亡者就被捕了。四个小孩都不允许走到有障碍物或其他人的方格上，也不能走出花园，因而，四个小孩总是会位于不同的方格上面。

这些小孩都是非常聪明的，三个追捕者也是团结一致。追捕者如果有可以捉到逃亡者的方法，那么他们就一定不会错过。逃亡者如果有不被捕获的方法，那么他也不会犯错。除此之外，追捕者会希望尽快地捉到逃亡者，而逃亡者即使在会被捕获的情况下也会尽可能地拖延时间。给定花园的障碍物的分布图和四个小孩的初始位置，你知道追捕者有方法捉到逃亡者吗？如果有，他们要经过多少轮后才能捉到逃亡者呢？

输入格式：

输入文件包含多组测试数据。每组测试数据第一行为两个整数 N 和 M ($1 \leq N \leq 10$, $1 \leq M \leq 10$)，为花园方格阵列的行数和列数。接下来 N 行，每行 M 个字符，可以为“.”、“#”、“0”和“X”，分别表示空地、障碍物、追捕者和逃亡者。追捕者总是会有三个，而且四个小孩一开始也都会在空地上面。

输出格式：

每组测试数据输出一行，若追捕者能够捉到逃亡者，则输出他们要经过多少轮后才能成功。轮数的计算包括追捕者和逃亡者进行行动的两轮，逃亡者被捕获的那一轮不算，因而结果总是一个奇数。具体输出格式请参考输出样例。

输入样例：

```
2 2
00
0X

3 3
000
##X
...

3 3
00#
###
.OX

3 4
00##
####
..OX

4 4
000.
....
....
...X

5 5
0...0
.....
..#.
.....
```


0...X
5 5
0...0
.....
...#.
.....
0...X
6 6
.....
.0..0.
..##..
..##..
.0..X.
.....
6 6
#.....
.0..0.
..##..
..##..
.0..X.
.....
10 10
.....
.....
..0....0..
.....
.....
.....
.....
..0....X..
.....
.....
10 10
.....
.#.#.#.#.#
..0....0.
.#.#.#.#.#
.....
.#.#.#.#.#
.....
.#.#.#.#.#
..0....X.
.#.#.#.#.#

输出样例：

```
The escapee will be captured after 1 steps
The escapee will be captured after 7 steps
The escapee will be captured after 5 steps
The escapee will never be captured
The escapee will be captured after 21 steps
The escapee will never be captured
The escapee will be captured after 41 steps
The escapee will never be captured
The escapee will be captured after 39 steps
The escapee will never be captured
The escapee will be captured after 51 steps
```

说明：

共有 5 个测试数据集，每个测试数据集为一个输入文件，包含多组测试数据。每个测试数据集从易到难分别为 5、10、15、30 和 40 分，对每个测试数据集分别执行一次程序，每次必须在运行时限 60 秒内结束程序并输出正确的答案才能得分。

所有数据均从标准输入设备（stdin/cin）读入，并写出到标准输出设备（stdout/cout）中。

五个测试数据集中输入 N、M 分别不大于 6、7、8、9 和 10。

2006 年百度之星程序设计大赛总决赛题目

俄罗斯方块 俄罗斯游戏中共有七种方块，每种方块都由四个方格组成，如下图所示，七种方块分别编号为 1~7。

游戏中，每次落下一个方块，落到一个宽度为 10 格的槽中。方块的下部一旦碰到槽的底部，或槽中已有的方块，就不能再移动。方块落下不动后，如果有某些行因落下的方块而填满，这些行将被消去。方块下落前，你可以控制方块的左右移动和旋转，以将其放在合适的位置。你对方块的所有移动和旋转操作在下落前（槽外）就计算完毕，然后直接下落到底，下落过程中不能再做操作。如果方块刚刚落下后顶部高度大于 17 行，游戏结束——即使此时有些行可以消除。

交互方式

你的程序应当包含 tetris_lib.h，并连接相应的库文件。库中的两个重要函数是：

```
void StartGame(int* t1, int* t2);
int Step(int r, int l, int* next);
```

你的程序应该首先调用 StartGame，其中 t1 和 t2 表示前两个方块的编号（t2 对应于传统游戏中的“下一个方块”）。接下来，你的程序每次可以使用 Step 函数下落一个方块，返回消去的行数。r 表示旋转方式（r=0, 1, 2, 3 分别表示顺时针旋转 0 度、90 度、180 度、270 度），l 表示方块在旋转后的最左边一格的列编号（从左到右依次为 1, 2, ..., 10），而 next 表示方块落下后新的下一个方块编号（0 代表没有下一个方块，下一次 Step 调用后库将自动终止你的程序）。你的程序不应自行终止。

关于自测的提示

调用 StartGame 函数时，库将从标准输入中读入若干行，每行包括一个整数，表示方块的编号。你可以利用这一点对你的程序进行测试。程序运行结束后，测试库将把结束原因和得分显示在标准输出中。库中还有两个函数可以用于自测：

```
void SetLog(const char* filename);
```

```
void Snapshot();
```

如果需要测试库记录程序的行为，请在调用 StartGame 之前调用 SetLog 函数。

评分规则

在与测试库交互的过程中，出现以下条件之一，则库将终止程序：

- 1、调用参数非法
- 2、方块刚刚落下后，其顶部高度大于 17 行
- 3、所有方块均已落下
- 4、本数据的运行总时间超过 1 秒钟

程序终止后，假设一次消去 1 行、2 行、3 行、4 行的次数分别为 a, b, c, d, 则该数据原始得分为 $2b+6c+10d$ 。

换句话说，消去单独的 1 行不得分。

对于每个数据，得分排名前八的程序分别得到 10, 7, 6, 5, 4, 3, 2, 1 分。如果原始得分相同，则消去行数多的排名在前；如果原始得分和消去行数都相同，则下落方块总数多的排名在前。

如果三者都相同，则得分相同。消去行数为 0 的程序不得分，即使它排在前八。

最终成绩按照 50 个数据的总分从大到小排序，如果总分相同则按所有数据的原始得分之和排序；如果仍有相同，则按所有数据的消去的总行数排序；如果仍相同，则名次相同。

百度的高级搜索方法(2007 年初赛)

题面描述：

你尝试过在百度上使用 site inurl 语法查询吗？如果还没有的话可以试一下：)

如输入 site:www.baidu.com inurl:news

则会搜出所有在 www.baidu.com 站点上的包含“news”子串的 url。

现在我们有两份数据，一份是 site_inurl.txt 一份是 url.txt

site_inurl.txt 中每行是一个 site inurl 语法组成的查询串，url.txt 中保存的是 url 列表。

你能否在 url 列表中找出所有能被 site_inurl.txt 中的查询串检索到的 url？

如 site_inurl.txt 内容如下：

```
site:www.baidu.com inurl:/more
```

```
site:zhidao.baidu.com inurl:/browse/
```

```
site:www.sina.com.cn inurl:www20041223am
```

url.txt 内容如下：

```
http://www.baidu.com/more/
```

```
http://www.baidu.com/guding/more.html
```

```
http://www.baidu.com/events/20060105/photomore.html
```

```
http://hi.baidu.com/browse/
```

```
http://hi.baidu.com/baidu/
```

```
http://www.sina.com.cn/head/www20021123am.shtml
```

```
http://www.sina.com.cn/head/www20041223am.shtml
```

则你的程序运行完输出的结果应该为：

```
http://www.baidu.com/more/
```

```
http://www.baidu.com/guding/more.html
```

<http://www.sina.com.cn/head/www20041223am.shtml>

程序以命令行形式传入这两个文件名，第一个参数为 site_inurl 文件对应的文件名，第二个参数为 url 列表对应的文件名，程序的输出请输出到标准输出。

Wii 游戏开始啦！（2007 年初赛）

题目描述

为了在紧张的上班时间内让员工们轻松些，百度休息室里放置着按摩椅、CD、高尔夫套装和 Wii 游戏机等休闲用品。其中最受欢迎的当然是游戏机。

wii 游戏机每个手柄需要使用两节电池（这两个电池可以是不同的品牌）。工程师们在玩游戏时。如果手柄没有电，他们都是将其中没电的电池拿走，并换上一个全新的电池，有电的必须继续使用。例如，已知三种电池的使用时间分别为 3 小时、5 小时和 8 小时。一开始，工程师使用 3 小时和 5 小时的电池。3 小时后，换上一个 8 小时的，再过 2 小时后，手柄再次没电时，已经没有电池可用了。

但如果一开始就使用那个 8 小时电量的电池，可以玩满 8 个小时。

告诉你每个品牌电池的使用时间以及该品牌电池的个数，请计算工程师们玩游戏时间的最小值和最大值。

输入格式

输入第一行为一个正整数 n ，表示电池的种数。接下来 n 行，每行两个整数 L 和 F ，表示使用时间为 L 的电池有 F 个。

输出格式

输出仅一行，包含两个整数，分别表示工程师们的最短游戏时间和最长游戏时间（短的时间在前）。两个整数之间以空格隔开。

输入样例

```
3
3 2
5 2
8 2
```

输出样例

```
5 8
```

实习生小胖的百度网页过滤器（2007 年初赛）

题目描述

百度网页采集器(Baiduspider)每天从互联网收录数亿网页，互联网的网页质量参差不齐。百度的工程师们每天都在改进方法来判断一个网页质量的好坏，使质量差的网页出现在检索结果中较后的位置。现在实习生小胖想到一个很简单的方法来判断一个网页内容的好坏，方法如下：

1. 利用数据挖掘技术在互联网语料库中挖掘出一批有特点的词汇，分为好词和坏词两种，好词标上正的权重，坏词标上负的权重；
2. 通过好词和坏词词典对每个网页计算网页总权重：从第一个字开始匹配，找到一个好词则加上相应的权重，找到一个坏词则减去相应的权重，下一次匹配将从找到的词末尾的下一个位置开始。
3. 坏词采用正向最短匹配：从当前匹配位置开始的若干连续汉字，如果形成多个坏词，则只计算最短的那个坏词的权重，下一次匹配将从这个最短坏词末尾的下一个位置开始。

4. 好词采取正向最长匹配：从当前匹配位置开始的若干连续汉字，如果形成多个“有效”好词，则只计算最长“有效”好词的权重，下一次匹配从这个最长“有效”好词末尾的下一个位置开始。

5. “无效”好词的定义：好词的一部分本身是一个坏词；或者好词的一部分与后续相邻的若干字组成一个坏词。

现在小胖已经做好了第 1 步的工作，有一个好词和坏词的列表（词典），但是由于没有对中文文本处理的程序经验，他想请未来的百度之星们帮他完成这个程序。

输入格式

输入第一行为一个字符串（网页正文）。从第二行开始为词典，格式为“词 空格 词的权重”。权重为一个带符号 32 位整数。如果权重为正，则为好词，反之则为坏词；不存在重复的词，不存在权重为 0 的词。测试数据中的词全部为 1-5 个字的中文，但作为“网页”的字符串中同时包含中文和 ASCII 字符，每个汉字占 2 个字节。并非“网页”中的所有字都在词典中。

样例输入

小胖之喷火龙骑士!!

小胖 6

喷火 -1

喷火龙 -1

火龙 -1

龙 4

龙骑 3

龙骑士 2

骑士 -2

士 3

输出格式

输出仅一行，为网页总权重（答案保证不超过带符号 32 位整数的范围）。

样例输出

7

样例解释

从“网页”中找到的好词为“小胖”和“龙”，坏词为“喷火”和“骑士”。特别要说明一下“龙”被识别为好词的原因——“喷火”和“喷火龙”均为坏词，按正向最短匹配得到“喷火”，接着往下匹配到好词“龙”、“龙骑”和“龙骑士”，但是由于“骑士”是坏词，所以“龙骑”、“龙骑士”无效而“龙”是最长的有效好词。注意题目描述中的匹配规则，好词的“有效”和“无效”只考虑该好词的一部分与后续字是否能够组成坏词，而不考虑和前面的字是否能够组成坏词——样例中的“龙”虽然可以与前面的字组成坏词“喷火龙”和“火龙”，但由于这两个词都是未能匹配成功的坏词，因此对好词“龙”的词性没有影响，可以累积“龙”的权重。

百度时间（2007 年初赛）

题目描述

Baidu 的服务器上使用的不是北京时间，而是 Baidu 时间。Baidu 时间的时分秒与北京时间相同，但是日期与北京时间不同，是用一个正整数表示从 2000 年 1 月 1 日起的第几天。

现在就请大家设计一个程序将北京时间转换为百度时间。

输入格式

输入数据的每一行为一个待转化的北京时间，格式包括两种：

一种为：YYYY-MM-DD，（YYYY 表示四位数年份，MM 为两位月份，DD 为两位日期）；

另一种为：MM/DD/YYYY，（YYYY 表示四位数年份，MM 为两位月份，DD 为两位日期）；
不符合任何一种格式的输入视为非法输入。

输出格式

每个数据输出一行。如果格式正确，输出一个正整数，否则输出 Error。

输入样例

2006-03-21

AStar2007

04/22/2007

输出样例

2149

Error

2463

SQL 中的 SELECT 语句（2007 年初赛）

题目描述：

SQL 中 SELECT 语句用于从数据库中查询记录。某个工程项目数据库中有一个表，表中满足某个条件的记录数要被频繁查询。查询表中满足某条件的记录数的 SELECT 语句如下：

SELECT 记录数语句：“SELECT COUNT(*)” + “ ” + FROM 部分+ “ ” +WHERE 部分

或者

“SELECT COUNT(*)” + “ ” + FROM 部分

SELECT 记录语句：“SELECT *” + “ ” + FROM 部分+ “ ” +WHERE 部分

或者

“SELECT *” + “ ” + FROM 部分

TABLENAME: 基本单元

FROM 部分：“FROM” + “ ” +TABLENAME 或者 “FROM (” +SELECT 记录语句+ “)”

PROPERTY: 基本单元

VALUE: 基本单元

条件部分： PROPERTY + “=” +VALUE 或者 PROPERTY + “=” +VALUE+ “ ” + “and” + “ ” +条件部分

WHERE 部分：“WHERE” + “ ” +条件部分

基本单元：由大小写字母和数字字符（0-9）组成的长度不超过 10 的非空字符串

给定表中的所有记录，根据给定的 SELECT 记录数语句要求输出相应的结果（满足条件的总记录数）。

数据输入：

有多个测试例子：

每个测试例子：第一行两个整数 N（ $2 < N \leq 1000$ ）和 M（ $0 < M \leq 10000$ ）分别表示数据库中表的记录数和查询次数；第二行是一个表名（TABLENAME）；第三行表中的 C 列属性名（PROPERTY）（ $0 < C < 11$ ），之间用空格隔开，列属性名各不相同；接下来 N 行，每行表示一个记录，有 C 个 VALUE，之间用空格隔开；接下去有 M 行，每行一个 SELECT 记录数语句，每个 SELECT 记录数语句长度（包括空格）不超过 1000，输入数据保证每条 SELECT 记录数语句满足上面的定义，并且 SELECT 记录数语句中出现的表名和输入的表名一致。

数据输出：

每个测试例子输出 M 行每行一个整数表示 SELECT 语句输出的结果。

输入样例：

5 6

```

Book
BookName Price PublishDate Author
NBAsports 10 2004 dearboy
SQL 20 2002 absorbed
IntrotoAlgorithm 59 2002 Thomas
MultipeView 60 2002 RichardHautley
NBAsports 10 2004 dearboy
SELECT COUNT(*) FROM Book WHERE BookName=NBAsports and Author=dearboy
SELECT COUNT(*) FROM Book WHERE Price=20
SELECT COUNT(*) FROM Book WHERE Author=lala
SELECT COUNT(*) FROM (SELECT * FROM Book WHERE BookName=NBAsports)
SELECT COUNT(*) FROM (SELECT * FROM Book WHERE BookName=NBAsports) WHERE Price=20
SELECT COUNT(*) FROM Book

```

输出样例:

```

2
1
0
2
0
5

```

注意: 大量数据建议使用 scanf(), printf(), gets() 读写数据

繁忙的会议室预定问题 (2007 年初赛)

题目描述

百度由最开始的 7 人团队迅速发展为几千人的大团队, 而工程师们经常需要在一起进行“头脑风暴”, 这样会议室就成了紧缺资源。为了有效利用资源, 大家决定制定规则, 自动安排会议室的使用。为了公平起见, 应按照申请时间从早到晚依次考虑, 先到先得, 且申请一旦被接受就不能取消。注意同一时间开的不同会议必须不同的会议室, 而同一个人不能同时参加两个会议。

输入格式

输入第一行为会议室总数 n 和请求总数 m ; 第二行是 n 个整数, 表示会议室能够容量的人数。以下 m 行每行是一个请求, 按请求时间先后顺序排列 (即应优先满足在输入中更早出现的请求)。

每个请求中第一个是整数, 表示会议需要的时间长度 (单位: 小时); 之后为与会人名单。人名由不超过四个汉字组成, 用半角逗号分隔 (每人名字固定且唯一, 有重名的也在登记时区分开)。名单后的数字表示可以安排会议的时间, 也以半角逗号分隔, 如 10, 11, 14, 15 表示第 10, 11, 14, 15 个小时可以开会 (会议时间为 9 到 19 之间的正整数)。

输出格式

输出 m 个数, 依次表示每个请求是否被接受。1 表示接受, 0 表示不接受。

输入样例:

```

4
20 2
3 张三, 李四, 王五 10, 11, 12, 14, 15
1 张三 12
4 王六, 王七, 王八, 王九, 王十 9, 10, 11, 12, 13, 14, 15

```

2 张三 14, 15

输出样例:

1 0 0 1

水果开会时段（2007 年初赛）

题目描述:

每个百度工程师团队都有一笔还算丰裕的食品经费，足够每天购置多种水果。水果往往下午送达公司前台。前台的姐姐们只要看到同时出现五种或以上的水果，就称之为“水果开会”。

从搜索引擎切词的语法角度，只要两种水果的名字中有一个字相同就属于同样的类别。例如“小雪梨”和“大雪梨”是同一种水果，而“核桃”和“水蜜桃”也被认为是同一种水果。尤其要指出的是，如果有三种水果 x , y , z 同时在前台出现，且 x 和 y 是同一种水果， y 和 z 也是同一种水果的时候， x 和 z 也被认为是同一种水果。现在前台的姐姐们想知道，今天是否有“水果开会”——五种或更多的水果同时在前台出现。

输入格式

输入的第一行只有一个整数 n ，表示购置水果的组数。接下来的 n 行表示水果的到达时间、取走时间（时间用 1200 到 1900 之间的正整数表示，保证取走时间大于到达时间）。剩下的字符串以空格分割每一种水果。如“1400 1600 雪梨 水蜜桃”，表示下午两点到四点（包含两点和四点这两个时间点），雪梨和水蜜桃会在前台等待开会。每种水果名称由不超过十个汉字组成。

输出格式

输出仅一行，包含一个字符串 Yes 或 No，分别表示今天水果开会与否。

输入样例 1

3

1200 1400 雪梨 柠檬

1300 1400 西瓜 苹果

1400 1800 花生 水蜜桃

输出样例 1

Yes

输入样例 2

3

1200 1400 雪梨 柠檬

1400 1500 哦 大梨 呀

1500 1800 咦 大梨

输出样例 2

No

大话西游与数字游戏（2007 年初赛）

题目描述

“叉烧鸡翅膀，我呀最爱吃！……”

百度 spider 组的“黑龙潭之行”在烤着鸡翅，唱着星爷的经典时达到高潮。大家在篝火旁围成一圈，开始玩“数 7”加强版游戏，规则如下：

规则 1：遇 7 的倍数或含 7 的数时 pass。

规则 2：遇有包含相同数字的数时 pass。注意相同数字不必相邻。例如 121。

数错的惩罚很残酷——吞食烤全羊。为避免惩罚，百度工程师们需要你——史上最强程序员的帮助。百度

工程师想知道：

req1 x: 符合规则 1 的第 x 个数是什么？

req2 y: 符合规则 2 的第 y 个数是什么？

req12 z: 同时符合规则 1、2 的第 z 个数是什么？

query n: 数 n 是规则 1 中的第几个数，是规则 2 中的第几个数？

输入格式

输入的每一行为一个查询，由一个查询词和一个无符号整型数组成。共有四种查询，查询词分别为 req1、req2、req12、query（区分大小写）。

输出格式

前三种查询输出一个无符号整型的解。对于“query n”的查询，若 n 是规则中的数则输出相应的解，否则输出-1。

输入样例

req1 10

req2 10

req12 10

query 14

输出样例

11

10

12

-1 13

补充说明

输入数据共分五组，前四组中： $1 \leq x \leq 10000000$, $1 \leq y \leq 1000000$, $1 \leq z \leq 250000$, $1 \leq n \leq 24000000$ ；第五组中的 y 可能达到 5000000

[2007 年百度之星程序设计大赛复赛题目](#)

好心的出租车司机

题目描述

北京的一位出租车司机向你抱怨：城市发展太快，公路越来越多，他已经疲于计算行驶路线，于是求助你开发一个自动导航的工具。

出租车只能在公路上行驶。所有的公路都是笔直、双向的，相交的公路视为连通（可以在交叉点处从一条公路开到另一公路上）。由于道路施工，整个城市的公路系统可能并不完全通畅。如果乘客的目的地不在公路边，则乘客下车后要步行前往，步行路线不受公路限制。这位好心的司机还特别提出，乘客步行距离越短越好；其次，出租车行驶里程越短越好。

方便起见，用笛卡尔坐标系来描述城市地图，所有坐标都在第一象限 $[0, 1000]$ 的范围内。公路宽度忽略不计。

输入格式

第一行是一个数字 k，代表公路条数。以下 k 行每行用 4 个实数描述一条公路（用空格隔开），前两个表示公路起点，后两个表示公路终点。下一行包含 4 个实数（用空格隔开），前两个表示乘客上车点，后两个表示乘客目的地坐标。不相交公路间的最短距离至少为 10^{-4} 。

输出格式

仅一行，为出租车行驶的里程数，保留一位小数。

输出格式

2

2.0 2.0 10.0 10.0

10.0 2.0 2.0 10.0

3.0 3.0 9.0 4.0

输出样例

7.8

评分方法

本题有 20 组数据，满足 $k \leq 100$ ，公路的交点数不超过 10000。

Robots.txt 协议

题目描述

搜索引擎是靠 Web Robot (又称 Spider) 来收集互联网上浩如烟海的网页的。Spider 就像一个旅行家一般，不知疲倦地奔波于万维网的空间，将遇到的页面收集下来供搜索引擎索引。对于一个网站的管理员来说，如果希望搜索引擎只收录自己指定的内容，或者指定某些不希望搜索引擎访问的内容，该如何去做呢？他需要的就是 Robots Exclusion Protocol 协议，这里简单的称它做 Robots.txt 协议。

Robots.txt 是一个放置在网站根目录下的纯文本文件。举例来说，当 Spider 访问一个网站（比如 <http://www.example.com>）时，首先会检查该网站中是否存在 <http://www.example.com/robots.txt> 这个文件，如果 Spider 找到这个文件，它会根据这个文件的内容，来确定它访问权限的范围。

www.robotstxt.org 是 robots.txt 协议的 Home Page，在这个站点上你可以找到很多 robots.txt 协议相关的资料。Robots.txt 协议在 <http://www.robotstxt.org/wc/norobots.html> 上有比较详尽的描述。你的任务就是编写 Spider 中的一个逻辑单元，这个单元的作用就是来判断一个网站的一些 URL 是否被禁止抓取。对方网站的站点在这里假设是 www.example.com，这个 Spider 的 User-agent 当然是 Baiduspider。注意，这个逻辑单元除了本身的判断逻辑要求与 robots.txt 协议一致外，还要注意容错的问题。互联网上纷繁芜杂，会出现很多意想不到的错误。如何能够对一个对错参半的 robots.txt 进行解析，把其中正确的挑拣出来、把错误的部分忽略掉，也是一个不小的挑战哦。都会遇到什么错误？在开始爬行互联网之前，谁都不知道。

输入格式

第一行是一个正整数 m ，表示 robots.txt 文件的行数，后面跟 m 行，是 robots.txt 的全文。下一行包含一个正整数 n ，表示 URL 的行数，后面跟 n 行 URL，这个就是你要判断的 URL 的列表。

输出格式

每条 URL 输出一行，每行两列，第一列是一个数字，如果这条 URL 被禁止，则输出 0，否则输出 1。第二列是这条 URL 本身。

输入样例

2

User-agent: *

Disallow: /tmp/

2

<http://www.example.com/index.html>

<http://www.example.com/tmp/somepage.html>

输出样例

```
1 http://www.example.com/index.html
0 http://www.example.com/tmp/somepage.html
```

评分方法

本题包含 20 组数据，均满足 $0 < n, m \leq 100$ 。

简单印象

题目描述

简单、可依赖是百度的性格，百度之星们又有怎样的性格呢？

有 n 名选手入围了百度之星程序设计大赛的复赛阶段。为了让选手相互之间有更多的了解和更好的交流，组委会工作人员邀请每位选手选择一些不同的词语来介绍自己的性格。每名选手需要为自己选定的每一个词语指定一个绝对值不超过 100 的整数权值 q ，表示这个词语在多大程度上描述了自己的性格。例如“美丽 90”表示该选手认为自己非常美丽，而“冷淡 -60”表示比较热情，而“外向 -5”表示稍微有一点点偏内向。你不需要考虑词语本身的语义，比如“外向 -5”不代表“内向 5”。

组委会发现不少选手的性格都有相似之处，所以希望找到一组词语尽量准确的描述所有的选手。对于选出的词语集合 S ，定义选手 i 被描述的精确程度 $P(i)$ 为 S 中所有词语在选手 i 眼中的权值之和（选手 i 没有提到的词权值视为 0），组委会希望让最小的 $P(i)$ 尽量大，因为这样才能让选出的词语能够描述所有选手，而不仅仅是部分。

另外，所选词语的个数也是有讲究的。词语太少会略显单薄，而词语太多又不方便宣传，所以组委会设定了词语个数的最小值 \min 和最大值 \max 。

输入数据

输入数据第一行为三个整数 n , \min 和 \max ，表示选手的个数、词语的最小值和最大值，接下来的 n 行，每行包括若干“词语-权值”对。词语保证由不超过 10 个汉字组成（用 GBK 编码，不含非汉字），权值为绝对值不超过 100 的整数。

输出数据

仅一行，输出各个词语，用空格隔开。次数总数必须不小于 \min 且不大于 \max ，每个词语都必须至少被一个选手提到过，否则视为非法输出。

输入样例

```
3 2 4
美丽 90 冷淡 -60 外向 -5 乐观 20
美丽 10 冷淡 20
外向 20 乐观 -5
```

输出样例

```
美丽 冷淡 外向
```

样例解释

如果把选手按照输入中出现的顺序编号为 $1 \sim 3$ ，则 $P(1)=90-60-5=25$ ， $P(2)=10+20=30$ ， $P(3)=20$ ，□□有 P 中的最小值为 20。

评分方法

本题包含 30 个测试点，每个测试点 10 分，共 300 分。

测试点 $1 \sim 10$ 满足 $n \leq 100$ ， $1 \leq \min \leq \max \leq 5$ ，涉及到的词语不超过 500 个。

测试点 $11 \sim 20$ 满足 $n \leq 200$ ， $10 \leq \min \leq \max \leq 15$ ，涉及到的词语不超过 2000 个。

测试点 21~30 满足 $n \leq 400$, $25 \leq \min \leq \max \leq 30$, 涉及到的词语不超过 10000 个。

每个测试点独立评分。对于每个测试点,非法输出的得分为 0,合法输出的得分大于 0。设合法输出的程序数为 M ,比程序 i 的方案严格更优的程序数为 $Y(i)$,则该测试点程序 i 的分值为 $10(1-Y(i)/M)$ 。换句话说,输出该测试点最优解的程序将获得 10 分,而最差解唯一的情况,输出最差解(但合法)的选手将得到 $10/M$ 分。注意:每个测试点的得分不必为整数。

紧急修复

背景

2050 年的一天,某市 k 家公司的计算机系统突然同时瘫痪。市政府很快找到了问题的所在,并开始研发一套自动修复整个城市的系统。自动修复系统启用后整个城市的计算机系统将恢复正常,但由于研发时间较长,在此之前各公司仍将蒙受不小的损失。为此,市政府决定派出 n 支维修能力相同的维修队到各公司进行手工修复,力图在自动修复系统启用前整个城市的总损失达到最小。

参数

城市被划分成 $R \times C$ 的网格,各行从上到下编号为 $1 \sim R$,各列从左到右编号为 $1 \sim C$ 。每个格子为空地、障碍物和建筑物之一(公司总是位于某个建筑物格子中)。维修队每次只能往上(行编号减 1)、下(行编号加 1)、左(列编号减 1)、右(列编号加 1)四个方向移动,第 i 个维修队每小时可以移动 $s(i)$ 格。维修队在任何时候都不能位于障碍物中,但建筑物可以从它上下左右的相邻空地进入,也可以从这些格子出去。注意:不能直接从一个建筑物格移动到另一个建筑物格,而必须先移动到相邻的空地,在空地上移动,然后再进入另一个建筑物。每个格子的实际尺寸很大,因此可以有多个维修队同时就在同一个格子中。第 i 家公司的位置为 $(r(i), c(i))$,瘫痪程度为 $B(i)$,每小时的经济损失为 $P(i)$ 元。瘫痪程度的单位是“队·小时”,即一支维修队每小时可以把一个公司的瘫痪程度降低 1,而如果 m 支维修队同时为一个公司修复,每小时可以把该公司的瘫痪程度降低 m 。

注意,在完全修复之前,每个小时的经济损失不变。

修复计划

政府的修复计划应包含每个小时各维修队所执行的命令。这些命令包括:

1. REST

该命令不进行任何操作。

2. MOVE

按照 seq 进行移动。其中 seq 为一个长度不超过 s 的非空字符串(如果不需要移动,请使用 REST 命令)。如果 seq 的长度超过 s ,则从第 $s+1$ 个字符开始的剩余部分将被删除;如果该命令不能完全成功的执行,则从第一个非法移动开始的所有后续移动将被忽略。

3. REPAIR

对当前公司进行维修。如果当前公司已经修复或者当前位置不是公司,该命令将被忽略。该命令后面加的所有参数将被忽略。

需要注意的是,每个小时只能执行一条指令,例如不能在 MOVE 之后立刻 REPAIR,必须等待下一个小时。

输入格式

输入的第一行为三个正整数 R, C, T 即网格的行数、列数和自动修复系统的启用时间。以下 R 行每行 C 个字符,表示该城市的地图。点表示空地, # 表示障碍物, 0 表示建筑物。

下一行包含一个正整数 k ,表示公司的数目。以下 k 行每行四个整数 r, c, B, P ,其中 (r, c) 表示该公司坐标($1 \leq r \leq R, 1 \leq c \leq C$), B 为该公司的初始瘫痪程度, P 为每小时的经济损失。 (r, c) 处保证为一个建筑物格。 B 和 P 保证大于 0。

下一行包含两个正整数 n, s ,表示维修队的个数和每小时最多移动的格子数。以下 n 行每行包含两个整数

r, c , 表示该维修队的初始位置。维修队按照输入文件中的顺序编号为 $1 \sim n$ 。

输出格式

输出每 n 行为一组, 描述一个小时各维修队的发出的命令。共 T 组, 一共 nT 行。所有格式非法的指令将被忽略 (等效于 REST)。尽管如此, 如果程序输出的命令中没有 REPAIR, 或者所有的 REPAIR 都没有成功执行, 则输出将视为非法。

输入样例

```
4 7 5
...#0##
#. ....#
0...##0
#. ....
2
1 5 3 5
3 7 4 6
3
4 7 5
1 1 5
3 1 5
```

输出样例

```
MOVE U
MOVE RRRD
MOVE RDRURD
REPAIR
MOVE DRRU
MOVE DRRRU
REPAIR
REPAIR
REPAIR
REPAIR
REPAIR
MOVE D
REST
REST
REPAIR
```

模拟器

每小时的模拟过程如下:

第一步: 对于每个尚未修复的公司 i , 将 $P(i)$ 累加进总损失。

第二步: 按照序列从小到大的顺序依次执行各维修队的指令。

为了更清晰的说明规则并帮助选手设计和测试程序, 命题组开发了一个简单的模拟器。该模拟器根据输入数据和程序输出进行模拟, 给出详细模拟过程, 以及最后的结果。最终的测试将严格按照该脚本的输出进行评判。模拟器用 python 编写, 没有安装 python 的选手可以在 <http://www.python.org> 下载最新版本。

评分方法

本题包含 30 个测试点, 每个测试点 10 分, 共 300 分。

测试点 $1 \sim 10$ 满足 $R, C \leq 10, n \leq 5, k \leq 10, B \leq 15, T \leq 30$

测试点 11~20 满足 $R, C \leq 30, n \leq 10, k \leq 20, B \leq 50, T \leq 500$

测试点 21~30 满足 $R, C \leq 100, n, k \leq 100, B \leq 1000, T \leq 10000$

每个测试点独立评分。对于每个测试点，非法输出的得分为 0，合法输出的得分大于 0。设合法输出的程序数为 M ，比程序 i 的方案严格更优的程序数为 $Y(i)$ ，则该测试点程序 i 的分值为 $10(1-Y(i)/M)$ 。换句话说，输出该测试点最优解的程序将获得 10 分，而最差解唯一的情况，输出最差解（但合法）的选手将得到 $10/M$ 分。注意：每个测试点的得分不必为整数。

[2007 年百度之星程序设计大赛总决赛题目](#)

比赛目标：

完成基于 CounterStrike 1.5 的 podbot AI，在 fy_iceworld_plus, de_dust2, de_inferno 三张地图上与对手 AI 展开较量。Podbot 的基本代码框架由平台给出，请选手自行参考研究。

有待完成的代码位于 bot_T.cpp bot_T.h bot_CT.cpp bot_CT.h 中，各个函数功能与接口请选手参考代码中的注释说明。

比赛方式：

1. 提交代码：请将 bot_T.cpp bot_T.h bot_CT.cpp bot_CT.h 压缩成一个 rar 压缩文件，通过比赛网站提交：

第一二组使用 <http://192.168.0.3/>

第三四组使用 <http://192.168.0.4/>

第五六组使用 <http://192.168.0.5/>

2. 赛制：

第一阶段：从比赛开始至之后 4 小时整。提交第一阶段 bot 代码用于 iceworld 测试，测试结果将决定第二阶段分组。

第二阶段：使用 dust2 和 inferno 两张地图，赛程是由小组循环决出前八。前八由淘汰赛决定胜负，判定最终名次。

操作说明

编译

进入目录 C:/astar2007/MSYS

双击运行 msys.bat

在 msys 界面中输入 `cd /c/astar2007/PODBOT`

执行命令 `make`，如果 `make` 成功，则在 obj.win32 子目录中有 podbot_mm.dll 生成

调试

1. 在代码中使用 UTIL_ServerPrint 可以向 console 打印信息在游戏中使用
2. 可以记录 log 文件以供分析检查之用

运行

1. 将 podbot_mm.dll 拷贝到 C:\Program Files\反恐精英中文站 CS1.5 中文硬盘版\CS1.5 中文硬盘版\cstrike\addons\podbot 覆盖原有文件即安装成功。
2. 双击桌面上的 cstrike 快捷方式，启动游戏，单击进入游戏，建立一个 11 人局域网游戏。进入游戏后按 6 进入观察模式
3. 控制 bot
 - (1) 按 = 7 杀掉所有自动 bot
 - (2) 按 = 5 添加 bot，选择 godlike 且 aggressive 的 bot
 - (3) 观察游戏结果 按 ~ 可以观察 Server 调试信息输出

规则说明：

0. 裁判组拥有最终裁定权

1. 胜负判定原则

(1) 地图为炸弹图时，如 de_dust2 和 de_inferno 时，以炸弹是否成功爆炸为胜负条件。当地图为 fy_iceworld_plus 时以杀伤数为胜负条件。

(2) 当由于赛程问题导致平局出现时，由裁判组判定

2. 违例判定原则

(1) AI 程序不正常导致程序无法运行时，经裁判组认定后判负

(2) 以各种形式 hack 对手数据，经裁判组认定后判负

2008 年百度之星程序设计大赛初赛第一场题目

1. 成语纠错（15 分）

问题背景

成语是中华民族的文化瑰宝，作为历史的缩影、智慧的结晶、汉语言的精华，闪烁着睿智的光芒。

你的任务是给一个错误的四字成语进行纠错，找到它的正确写法。具体来说，你只允许修改四个汉字中的其中一个，使得修改后的成语在给定的成语列表中出现。原先的错误成语保证不在成语列表中出现。

有时，这样的“纠错”结果并不惟一。例如“一糯千金”可以改为“一字千金”也可以改成“一诺千金”。但由于“糯”和“诺”是同音字，“一糯千金”实为“一诺千金”的可能性比较大。

因此，我们还将提供一个汉字分类表，要求修改前后的两个字必须属于同一个分类。

在这样的限制下，我们保证成语纠错的结果惟一。

注意

1、汉字均采用 GBK 编码(参见 FAQ)

2、每个汉字分类至少包含两个汉字，同一个汉字可能出现在多个类别中。

3、成语列表中的成语都是真实存在的四字成语。成语列表和待纠错成语中的所有汉字均在汉字分类表中的至少一个分类中出现。

输入格式

输入第一行包含两个整数 n, m ($1 \leq n \leq 200, 1 \leq m \leq 20000$)。 n 表示汉字类别的个数， m 表示成语的个数。

以下 n 行每行用一个无空白分隔符（空格、TAB）的汉字串表示一个分类中的所有汉字。注意，该汉字串最多可能包含 200 个汉字。

以下 m 行为成语列表，每行一个成语，恰好四个汉字。

最后一行为待纠错的成语，恰好四个汉字，且不在成语列表中出现。

输出格式

仅一行，为一个四字成语。在“修改必须在同一分类中进行”的限制下，输入数据保证纠错结果惟一。

样例输入

73

糯诺挪喏儒

字自子紫籽

前钱千牵浅

进近今仅紧金斤尽劲

完万

水睡税

山闪衫善扇杉

一诺千金

一字千金

万水千山

一糯千金

样例输出

一诺千金

2. 圆内五角星（20 分）

问题背景

如图，一个半径为 1 的圆周上有 5 个点。按角度制给出 5 个点的极角 A_i ($0 \leq A_i < 360$, $i=1..5$)。按下图的方法连成一个五角星, 计算圆被切割成的 11 个部分面积的方差。

具体地说, 假定 11 个区域的面积分别为 S_1, S_2, \dots, S_{11} , 那么面积的均值计算方法为:

$$M = (S_1 + S_2 + \dots + S_{11}) / 11$$

面积的方差计算方法为:

$$D = ((S_1 - M)^2 + (S_2 - M)^2 + \dots + (S_{11} - M)^2) / 11$$

输入格式

输入仅一行, 包含 5 个 $[0, 359]$ 内的互不相等的整数。

输出格式

输出仅一行, 包含一个实数, 即各部分面积的方差。输出保留小数点后 4 位。

样例输入

014472288216

样例 0.0647

3. 传输方案规划（30 分）

问题背景

$$\frac{\sum_{i=1}^n \sum_{j=1}^m T_{i,j} V_{i,j}}{\sum_{i=1}^n \sum_{j=1}^m T_{i,j} C_{i,j}}$$

面对艰巨复杂的技术挑战，百度所崇尚的系统设计哲学是“简单可依赖”，而百度的工程师们正在互联网世界中实践着这种理念。这里正好有一个挑战，让作为百度之星的你小试牛刀：

在处理数以百亿计的网络信息的过程中，有一个很常见的问题：怎么样将一个集群上的信息以最低的成本传输到另外一个集群上？

- 数据源集群 A 有 n 台服务器，编号为 $1, 2, \dots, n$ ， i 号服务器上待传输的数据量为 A_i ，单位是 GB。
- 目的地集群 B 有 m 台服务器，编号为 $1, 2, \dots, m$ ， j 号服务器上的空闲容量为 B_j ，单位为 GB。
- A 集群的 i 号服务器上的每 GB 数据对于 B 的集群的 j 号服务器收益为 $V_{i,j}$ ，从 A 集群的 i 号服务器向 B 集群的 j 号服务器传输 1GB 数据的开销为 $C_{i,j}$ 。

你的任务是在保证 A 中的所有数据传输完毕的前提下，性价比 V/C 尽量高。其中 V 为所有数据在 B 集群上的价值之和， C 为总开销。换句话说，若 A 集群的 i 号服务器向 B 集群的 j 号服务器发送了 $T_{i,j}$ 个 GB 的数据，则性价比定义为：

输入格式

第 1 行两个整数 n, m ($1 \leq n, m \leq 50$)，即集群 A 和 B 各自的服务器台数。

第 2 行包含 n 个不超过 100 的正整数 A_1, A_2, \dots, A_n ，即集群 A 中每台服务器的待传输数据量（单位：GB）。

第 3 行包含 m 个不超过 100 的正整数 B_1, B_2, \dots, B_m ，即集群 B 中每台服务器所能接受的最大数据量（单位：GB）。

第 4~ $n+3$ 行每行包含 m 个不超过 100 的非负整数 $V_{i,j}$ ，表示集群 A 的 i 号服务器中每 GB 数据对于集群 B 中的 j 号服务器的价值。

第 $n+4$ ~ $2n+3$ 行每行包含 m 个不超过 100 的正整数 $C_{i,j}$ ，表示集群 A 的 i 号服务器中每 GB 数据传输到集群 B 中的 j 号服务器所需要的开销。

输出格式

仅一行，为最大性价比。输出保留三位小数（四舍五入）。如果 A 的数据无法传输完毕，输出 -1。

样例输入

22
12
21
110
75
61
32

样例输出

2.091

样例解释

一个方案是：

集群 A 的 1 号服务器把所有数据传输到集群 B 的 1 号服务器，价值 11，开销 6。

集群 A 的 2 号服务器把 1GB 数据传输到集群 B 的 1 号服务器，价值 7，开销 3，然后把剩下的 1GB 数据传输到集群 B 的 2 号服务器，价值 5，开销 2。

性价比： $(11+7+5)/(6+3+2)=2.091$

另一个方案是：

集群 A 的 1 号服务器把所有数据传输到集群 B 的 2 号服务器，价值 0，开销 1。

集群 A 的 2 号服务器把所有数据传输到集群 B 的 1 号服务器，价值 14，开销 6。

性价比： $(0+14)/(1+6)=2$ 。

第一种方案更优。

4. 公平数（35 分）

问题背景

如果一个整数的十六进制表示（不含前导 0）中，前一半数字之和等于后一半数字之和，我们称它为公平数。

注意，如果该数的十六进制表示中包含奇数个数字，则正中间的数字既不属于前一半，又不属于后一半。

例如在十六进制下 $1D=7+7$ ，因此 1DE77 是公平数。数字 E 并不参与计算。

再例如，所有单个数字的十六进制数（即 0~F）均为公平数，但 F0 不是（不能把 F0 补充前导 0 写成 0F0，进而认为它是公平数）。

给出十六进制数 X, Y, K 和十六进制数字集合 S，求区间 [X, Y] 之内，有多少个公平数满足：

- 十六进制表达式（不包含前导 0）中每个数字均在集合 S 中
- 并且为 K 的倍数

输入格式

输入第一行为数字集 S，包含 0~9 以及大写字母 A~F。

每个数字或字母最多出现一次。

第二行包含 3 个十六进制正整数 K, X, Y，均不超过 10 个数字。

输出格式

仅一行，包含一个整数，即满足条件的公平数个数。

样例输入

124C

5100FFF

样例输出

4

样例解释

只有四个数满足条件：212，424，4C4，C1C。

[2008 年百度之星程序设计大赛初赛第二场题目](#)

1. 广告排名区间（10 分）

问题背景

shifen 广告消费预估系统可以估计出一段时间内一个特定的广告在检索结果中排在各个位置的几率。比如系统对某广告的输出如下：

$p_1=0.03, p_2=0.08, p_3=0.04, \dots$

这说明该广告展现在第 1 位的概率是 3%，展现在第 2 位的概率是 8%，展现在第 3 位的概率是 4%……

问题是：如何给出一个排名估计区间 $[i, j]$ ，使得广告出现在该区间中的概率大于或等于一个预设值 p ，同时这个区间所包含的元素尽可能的少。也可用数学语言来描述：给定数 p 和数列 p_1, p_2, \dots, p_n ，求 i 和 j ($1 \leq i \leq j \leq n$)，在满足 $p_i + p_{i+1} + \dots + p_j \geq p$ 的前提下让 $j-i$ 最小。

一般来说， p_i 只需保留 6 位小数就足够了。这样，若令 $a_i = 10^6 p_i$ ， $a = 10^6 p$ ，则 a 和所有的 a_i 均为 $[0, 10^6]$ 之间的整数。这样就避免了对实数的处理。

输入格式

第一行包含一个整数 n ($1 \leq n \leq 100,000$)。

以下 n 行每行包含一个 $[0, 10^6]$ 内的整数，依次为 a_1, a_2, \dots, a_n 。这 n 个整数之和保证不超过 10^6 。

最后一行包含一个 $[0, 10^6]$ 内的整数 a 。保证所有 a_i 之和不小于 a 。

输出格式

输出仅一行，包含一个整数，即 $j-i$ 的最小值。

样例输入

```
7
5
8
4
7
10
5
2
18
```

样例输出

```
2
```

样例解释

$a_2=8, a_3=4, a_4=7$ 之和为 19，满足条件。而任何两个相邻数之和均小于 18。

2. LZW 网页判重 (20 分)

问题背景

有一种简单的网页判重的方法，通过求两个网页内容的最长公共子序列 (LCS) 长度来判定两个网页的相似程度。如：

(网页 A) 老师：请用“果然”造句。

(网页 B) 学生：先吃水果，然后喝汽水……

它们的最长公共子序列为“果然”，长度为 2。注意这里的“子序列”并不要求连续。

类似的，下面两个网页：

(网页 A) 老师：请用“果然”造句。

(网页 B) 学生：先吃水果，然后喝汽水，果然拉肚子……

最长公共子序列还是“果然”，长度为 2。但不难看出，由于“果然”两个字在网页 B 中也曾连续出现，第二组网页比第一组更加“相似”。为了区分开这两种情况的区分度，我们改用一种称为 LZW 的理论。为了严格的叙述相似度的计算方法，我们首先定义“文本单元”。

假定网页用一个不包含空白字符（空格、回车换行、水平制表符）的字符串来表示。它只包含纯文本，没有标签。在计算相似度之前，你应该首先对该字符串进行处理，划分成一个个“文本单元”。每个文本单

位可以是一个中文字、英文单词（由一个或多个连续的半角英文字母和数字组成，正规表达式为 $[a-zA-Z0-9]^+$ ）、或者一个标点符号。

根据上述定义，同一个标点符号的全角和半角应该被作为不同的文本单元，尽管他们看起来可能很相近；每个单独全角英文和全角数字都应该被看成一个单独的文本单元，而连续的半角英文字母和数字应被看成一个整体。总之，全角的字符可以与中文字同等待待。

这样，网页被看成文本单元序列。例如，网页“内容? 1 2 345 6 ??web2.00#”切分出的文本单元序列为（为了显示方便，用下划线分隔各文本单元）：

内_容_? _1 _2 _345 _6 _? _? _web2 _._00 _#

而网页“why 内容相似??1234567890,web#00”的切分结果为：

why_内_容_相_似_? _? _1234567890 _._web_#_00

黑体部分给出了两个网页的一个公共子序列。注意“内容”、“??”分别在两个网页中都是连续出现的文本单元。为了奖励这种情况，LZW 规定一段由连续 k 个文本单元组成的字符串权值为 k^2 。在刚才的例子中，“内容”、“??”的权值均为 4。但“00”是一个数字串，应当被看成一个单独的文本单元。所以权值仅为 1。

根据上述规则，公共子序列“内容??00”的权值为 $2^2+2^2+1=9$ 。在所有可能的子序列中，这个权值是最大的。

给定两个网页，求他们的 LZW 相似度，即所有可能的公共子序列中的最大权值。

注意

1) 输入的网页内容以 GBK 编码(参见 FAQ)

2) 除了大小写英文字母和数字之外的其他半角字符均视为标点符号。

输入格式

包含两行，分别是网页 A 和 B 对应的字符串（不包含空白字符）。每行至少包含 5 个字节，最多包含 200 个字节。

输出格式

输出仅一行，包含一个整数，为两个网页的 LZW 相似度。

样例输入

内容? 123456??web2.00#

why 内容相似??1234567890,web#00

样例输出

9

样例解释

尽管两个网页里看上去都有“123456”但一方面第一个网页中混杂的全角和半角字符，而另一方面，即使全部改成半角字符，由于数字串“123456”和“1234567890”将分别看成一个单独的文本单元，因此无法部分匹配。

3. 钉子与木板（30 分）

问题背景

墙上有 n 个钉子，编号为 $1, 2, \dots, n$ 。其中钉子 i 的横坐标为 i ，纵坐标初始为 x_i 。可以进行两种操作：

0kv: 竖直移动钉子 k ，坐标变为 (k, v) 。

1stv: 若在高度为 v 处放一块横坐标范围是 $[s, t]$ 的水平木板，它将下落到什么高度？换句话说，求出钉子 $s, s+1, s+2, \dots, t$ 的纵坐标中，不超过 v 的最大值。如果这些钉子的高度全部大于 v ，则木板将落到地上，高度为 0。注意，在 T 操作时，水平木板只是用来测试的“临时木板”，将在测试后立即被拿走，不会影响到后续测试工作。

输入格式

第一行包含两个整数 n, m ，即钉子的个数和操作的个数 ($1 \leq n, m \leq 105$)。以下 n 行一个不超过 109 的非负整数，即 x_i 。

输出格式

按照输入的顺序，对于每个 T 操作输出一个整数，即该测试水平木板的最后高度。

样例输入

```
54
1
3
5
7
9
1246
0310
1357
1355
```

样例输出

```
5
7
0
```

4. 圆面覆盖 (40 分)

问题背景

在平面上有一个长为 L ，宽为 W 的长方形，左下角坐标为 $(0, 0)$ ，右上角坐标为 (L, W) 。给定一些圆，第 i 个圆的圆心坐标为 (x_i, y_i) ，半径为 R_i 。

你的任务是求最小的正实数 k ，使得把每个圆的半径变为原来的 k 倍后（即：第 i 个圆半径变为 kR_i ，圆心位置不变），长方形将被这些圆完全覆盖。换句话说，长方形内部或边界上的任意点均至少在一个圆的内部或边界上。

输入格式

输入第一行包含三个整数 n, L, W ($1 \leq n \leq 50, 1 \leq L, W \leq 1000$)，即圆的个数、长方形的长和宽。

以下 n 行，每行三个不超过 1000 的正整数 x_i, y_i 和 R_i 。

输出格式

仅一行，包含一个实数 k ，保留小数点后三位。

样例输入

```
122
111
```

样例输出

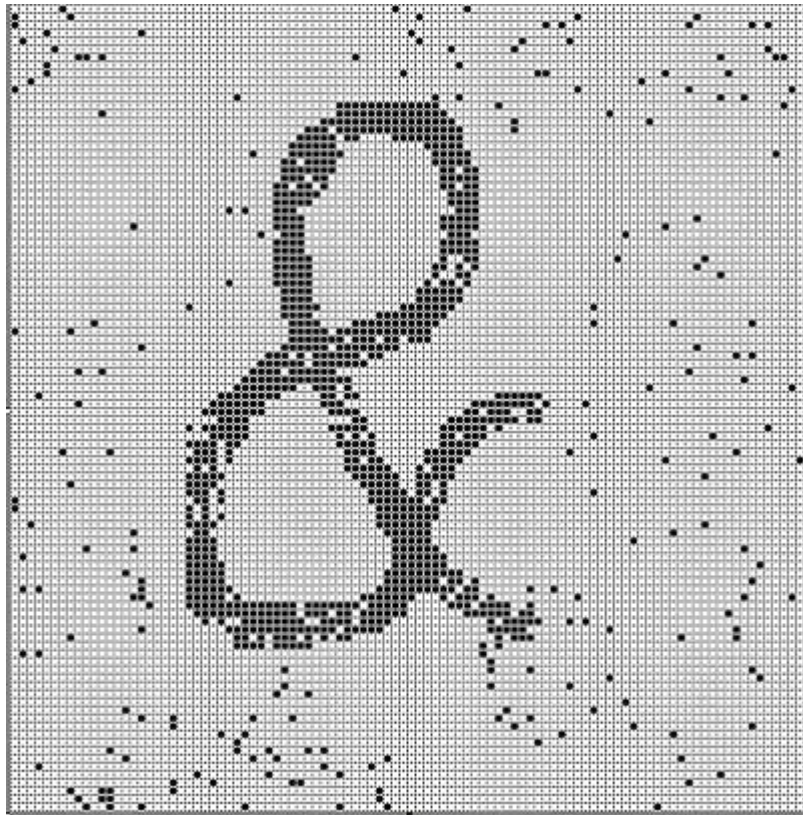
```
1.414
```

[08 年百度之星程序设计大赛复赛题目](#)

1. 验证码识别

问题背景

Baidu 的论坛抓取机器人小 A，最近的抓取表现越来越差了。工程师小明发现原来是很多论坛需要注册才能浏览，而且在注册的时候需要填验证码。如图所示



于是小明准备升级小 A 机器人，让它能够自动识别验证码。你能帮助小明设计识别验证码的程序吗？为了控制开发难度，这个版本只需要识别！@#¥%&*+九个符号即可。我们已经为你将图片拆分成 100*100 个点的位图，每个位图只包含一个符号，如下图所示的符号&。

同时为了让这个过程更有趣一点，我们将程序设计成交互式的，即你的程序向测试程序提问，通过测试程序的回答收集信息，当信息足够的时候输出解答即可。

例如：（注意 “_” 代表下划线，而不是空格）

1. 提问:你的程序向 stdout 输出字符串 Q_1_3\n，代表查询坐标（1，3）点的黑白信息；
2. 回答:测试程序向你的程序的 stdin 写入 P_0\n，代表（1，3）点的颜色为白，同理，如果你的程序读到 P_1\n，代表（1，3）点的颜色为黑；
3. 1 和 2 步骤不断循环，经过若干次交互，你的程序已经找到了答案，则可以输出结果：你的程序向 stdout 输出 R_&\n，测试程序就会记录识别结果和询问次数并退出测试。

测试程序

我们准备了一个帮助你测试的客户端程序，点击下载 [Test.zip](#)，需要的 jre 版本是 1.6.0_05，下载 jre。

注意

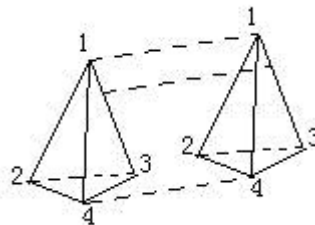
1. 识别单个图片，询问次数超过 2500 次则不得分，识别正确，且询问次数分数不高于 500 次得全分，高于 500 次后分数线性递减；
2. 识别单个图片，交互总时间超过 15s 则不得分；
3. 识别结果正确得 30% 的分数，识别结果错误不得任何分数；
4. 所有测试图形都由同一个出题人书写，字体方向正放向上；
5. 尺度不小于 50 像素，即主体符号无噪声包围盒的长宽不同时小于 50 像素；
6. 图片随机噪声点比例不高于 15%，图片上的黑白点都有可能随机翻转，而在添加噪声的过程中我们已经保证主体符号的含义是确定的；
7. Test. zip 中的测试数据是无噪声的，与评测数据不同，请特别注意。

2. 寻找所有的同构象（图片及公式暂缺）

问题背景

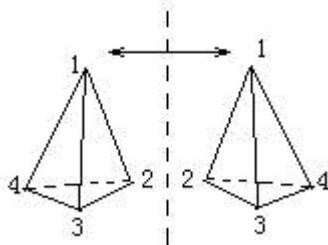
三维空间中，为一个物体定义如下三种基本的变换：

平移：物体沿着一条直线 α 进行一段位移 δ 。对于物体中的任意一点，其起始位置和终止位置连成的线

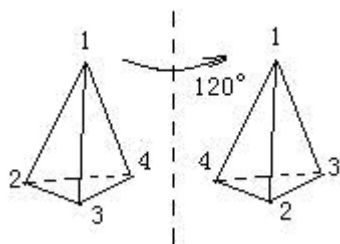


段平行于 α ，长度为 δ 。

反射：物体相对于一个平面 β 作镜像。对于物体中的任意一点，其起始位置和终止位置连成的线段垂直于 β ，并且到 β 的距离相等。



旋转：物体绕着一条轴线 γ 转过一定角度 ϕ 。对于物体中的任意一点，由其起始位置和终止位置对 γ 作垂线，垂足重合，垂线段长度相等，并且两条垂线所成的角度为 ϕ 。



物体 A 经过上述三种基本变换或基本变换的有限次组合，变成物体 A' ，若 A' 与 A 完全重合（对于空间中的任意一点，如果它属于 A' ，则一定属于 A ，反之亦然），则称 A' 为 A 的一个同构象。

需要注意的是，上面所说的物体和它的同构象并不一定是相同的物体，可以理解为物体上的每个点都是“有记号”的，虽然二者在形状、体积、位置上完全重合，但其中只要有一个点经过变换以后位置发生了改变，就是不同的物体。例如，线段 u 关于它的中垂面做反射得到线段 v ，虽然 u 和 v 在三维空间中重合，但 u 上的点变换到 v 以后位置发生了改变，所以 u 和 v 是不同的物体。

从定义出发，容易证明，同构象满足下面的 3 条性质：

1. 自反性： A 是 A 自身的一个同构象。
2. 对称性：若 B 是 A 的一个同构象，则 A 是 B 的一个同构象。
3. 传递性：若 B 是 A 的一个同构象， C 是 B 的一个同构象，则 C 是 A 的一个同构象。

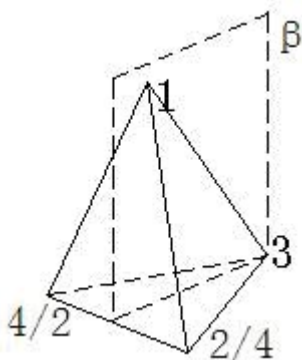
推论：若 B 是 A 的一个同构象， C 也是 A 的一个同构象，则 B 和 C 彼此互为同构象。

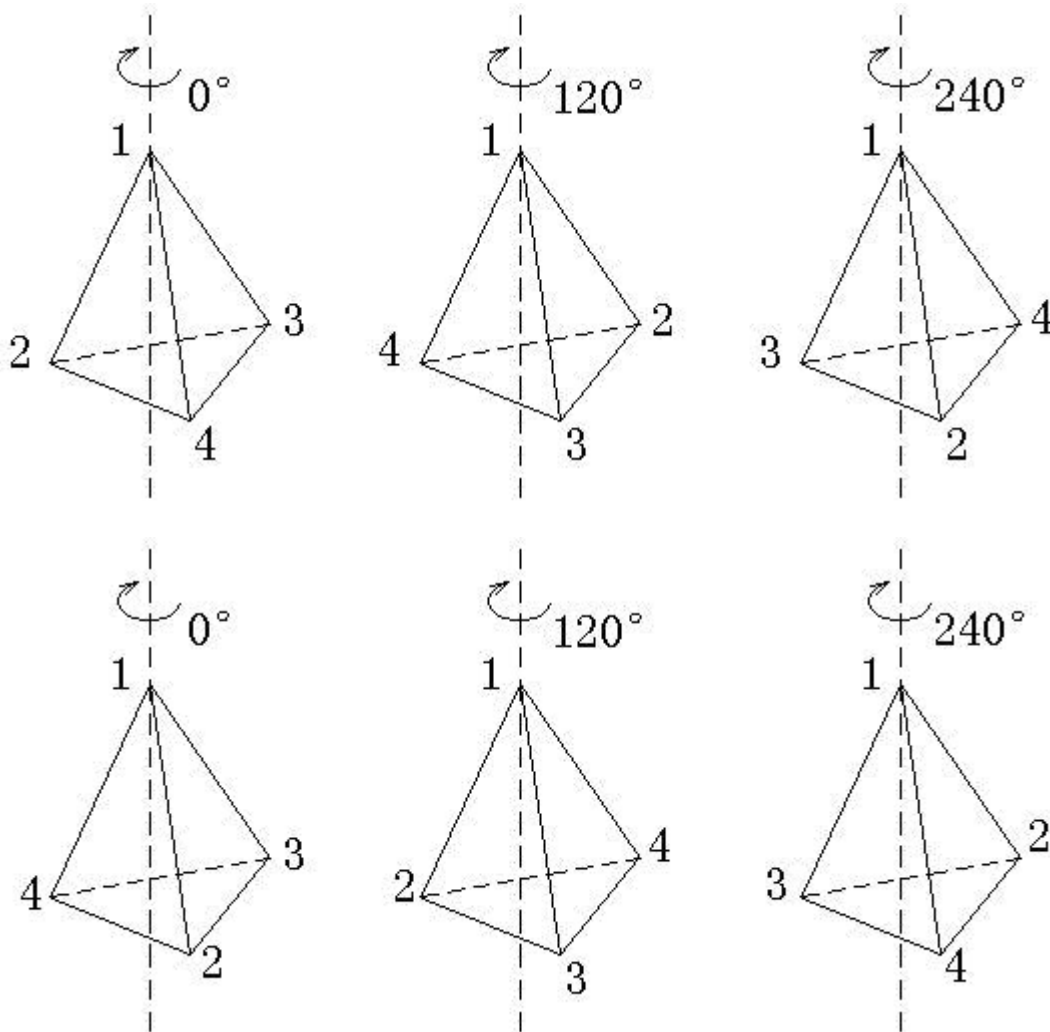
B 和 C 都是 A 的同构象，若有一个基本变换序列 S 把 B 变换到 C ，且满足对于任意点 p 属于 B ， S 把 p 变换到 p 自身，则称 B 和 C 为相同的同构象。例如，线段 u 绕着中垂线旋转 180° 得到一个同构象 v ，还可以关于中垂面反射得到一个同构象 v' ， v 和 v' 就是相同的同构象。

同构象的相同和不同是对本题非常重要的概念。事实上，由于 B 和 C 都是 A 的同构象，由上面的推论， B 和 C 一定互为同构象，因此 S 是一定存在的。那么 B 和 C 相同的唯一要求就是 S 把 B 变换到 B 自身，其实同构象“相同”的本质就是 $B=C$ 。

A 的全部不同的同构象组成的集合叫做 A 的同构象集。例如，上面所举的线段的例子中， u 的同构象集就是 $\{u, v\}$ 。

又例如，一个正三棱锥的同构象集共有 6 个元素，分别沿中轴旋转 0° 、 120° 、 240° 得到其中 3 个，再沿图中的 β 平面做反射以后，同样旋转又得到另外 3 个，这 6 个同构象集中的元素任意两者都不相同：





题目输入一个凸多面体的所有顶点坐标，要求输出这个凸多面体的同构象集所包含的元素个数。

输入的点至少有 4 个，每个输入作为一个顶点可以唯一确定一个多面体。不需要考虑非凸的情况，不需要考虑多点共线的情况，也不要求处理所有点退化到一个平面多边形的情况。但是有可能存在 4 个或以上的点共面。

多点共面的判断规则如下：四个点组成的四面体中，一个面积最大的表面的面积为 S_1 ，三个面积较小的表面的面积之和为 S_2 ，若 $S_1 \geq S_2$ ，则可认为四点共面。若有更多的点共面，输入保证其中任意四个点都满足上述不等式。

如有需要，可使用三角形面积的 Heron 公式，这个公式是数值稳定的：

$$A = \frac{1}{4} \sqrt{(a + (b + c))(c - (a - b))(c + (a - b))(a + (b - c))}.$$

其中 a 、 b 、 c 分别为三角形三边长，且。

输入格式

输入第一行为一个整数，表示凸多面体的顶点数 n ，此后每行为三个浮点数，精度为小数点以后 10 位有效数字，表示一个顶点的直角坐标 (x, y, z) ，顶点可能按照任意的顺序给出。

输入顶点数目； n ，均小于 10000。

输出格式

输出只有一行，为一个整数，表示同构象集元素的个数。

样例输入

输入每行为一个顶点的直角坐标 (x, y, z) ，精度为小数点以后 10 位有效数字：

```
4
0.00000000000.00000000000.00000000000
1.00000000000.00000000000.00000000000
0.50000000000.86602540380.00000000000
0.50000000000.288675134620.00000000000
```

样例输出

```
6
```

样例解释

样例输入对应一个正三棱锥，前三个点是底面正三角形，位于 $z=0$ 平面上，第四个点是正三棱锥的顶点，这个三棱锥的高是 20，形状比较细长。根据前面所举的例子，一个正三棱锥的同构象集有六个元素，因此输出为 6。

3. 黑白树

问题背景

在图论中，包含 n 个结点（结点编号为 $1 \sim n$ ）、 $n-1$ 条边的无向连通图被称为树。在树中，任意一对结点间的简单路径总是惟一的。你拥有一棵白色的树——所有节点都是白色的。接下来，你需要处理 c 条指令：

1. 修改指令 (0v)：改变一个给定结点的颜色（白变黑，黑变白）；
2. 查询指令 (1v)：询问从结点 1 到一个给定结点所经过的第一个黑色结点编号（假设沿着简单路径走）。注意，在查询指令中，必须从结点 1 而不是结点 v 出发。如果结点 1 本身就是黑色，查询指令应该返回 1。

输入说明

第一行包含两个正整数 n, c ，即结点数和指令条数。以下 $n-1$ 行，每行两个正整数 (u_i, v_i) ($1 \leq u_i < v_i \leq n$)，表示结点 u_i 到 v_i 之间有一条无向边。以下 C 行，每行两个整数 (C, V) 。当 $C=0$ 时表示修改指令，其中 V 代表被修改的结点编号； $C=1$ 时表示查询指令。你的程序需要输出结点 1 到结点 V 之间路径的第一个黑色结点编号。在第一条指令执行前，所有结点都是白色的。

输出格式

对于每个查询操作 ($c=1$ 的操作)，输出一行，包含一个整数，即第一个黑色结点的编号。如果不存在黑色结点，输出 -1。

样例输入

```
98
12
13
24
29
59
79
89
68
13
08
16
17
```

02

19

02

19

样例输出

-1

8

-1

2

-1

评分说明

共有 30 个数据，分为 3 组，按组计分。这三组的满分分别为 20, 30, 50 分。

第一组: $n=5000, m=4000000$

第二组: $n=100000, m=3000000$

第三组: $n=1000000, m=1000000$

每组数据中只要有一个数据运行超过 5 秒或者答案错，该组计 0 分。

否则，该组数据的得分取决于其中运行时间最长的数据的运行时间：运行时间越短，得分越高。

4. 度度熊与西瓜

问题背景

和其它熊不同，度度熊最喜欢吃的东西是西瓜，但是西瓜有一个很让它讨厌的地方——有很多的籽儿。可怜的度度熊并不知道这个世界上原来还有无籽西瓜这个东西。度度熊是一只很懒很懒的熊，并且它以 LarryWall 的名言作为自勉——懒是一种美德。它吃西瓜从不吐籽儿——因为太麻烦了。当然，这种美德是有代价的，度度熊经常因为吃西瓜籽闹肚子。

有一天，这只懒熊突发奇想——如果能用一刀把那些籽儿全部切掉该多爽啊（当然，秉承懒的美德，切两刀是很辛苦的）。度度熊有一个很奇怪的仪器，可以检测出西瓜籽儿的坐标位置。于是度度熊立马动起手来（对于这种事，它一点都不懒）。但是它很快就发现，如果刻意追求切出的西瓜块儿不带籽儿，那它能吃到的西瓜就只有很少的一点了，这真是一件很扫兴的事。于是，它就只好作了个让步，只要切剩下的西瓜部分包含的籽儿的个数不大于给定的值 m 就可以接受，这个 m 究竟是多少要取决于度度熊的心情和食欲。

可是度度熊的几何学得太差了，面对那些坐标它不知该用哪个公式去算，它只好求助于你——聪明的程序员了。

度度熊吃的西瓜的形状比较特殊，它看起来像这样子：

这看起来像个扁平的扇形，为了减轻你的负担，你可以将这个问题近似成一个平面的问题。西瓜是一个圆面的一部分，并且面积严格小于整个圆面的一半，最上面的顶点就是圆心。

给出西瓜的形状和籽儿的坐标，你的任务是求一个最佳的切割位置（刀面是直的），使得切出来的一块西瓜包含的籽儿不大于给定的数 m ，含不含有西瓜皮都没有关系——度度熊不讨厌吃西瓜皮。在这样的条件下当然是要求切出来的部分尽可能大了（度度熊现在很饿）。由于已简化为平面问题，所以你的任务是输出吃到的部分的面积。

注意：你可以忽略西瓜皮和西瓜籽的大小。

所有的坐标以极坐标格式输入。具体的格式是 (ρ, θ) ，分别表示（极径，极角），极角以角度为单位，

圆心在极点。

其中 $0 < \rho, 0 \leq \theta < 360$

名词解释

极坐标

在平面内取一个定点 O，叫极点，引一条射线 Ox，叫做极轴，再选定一个长度单位和角度的正方向(本题取逆时针方向)。对于平面内任何一点 M，用 ρ 表示线段 OM 的长度， θ 表示从 Ox 到 OM 的角度， ρ 叫做点 M 的极径， θ 叫做点 M 的极角，有序数对 (ρ, θ) 就叫点 M 的极坐标，这样建立的坐标系叫做极坐标系。

输入格式

第一行一个正整数 T，说明共 T 组数据， $T \leq 10$ 。

第二行一个正整数 r，说明西瓜的半径是 r， $r \leq 106$ 。

第三行一个非负整数 m，说明度度熊能够忍受的西瓜籽的个数为 m， $m \leq 1000$ 。

第四行两个非负整数 a, b，分别是西瓜皮的两个端点的极角， $0 \leq a < b < 360, 0 < b - a$

接下来的 n 行表示 n 个籽儿的坐标。每行两个整数表示(极径，极角)，籽儿保证在西瓜内，但可能在边界上。

输出格式

对每一组数据，输出单独一行，表示度度熊能切出来 " 包含不大于 m 个籽儿的西瓜片 " 的最大面积(四舍五入到小数点后 5 位)。

样例输入

```
2
10
1
090
2
145
745
10
0
090
2
145
745
```

样例输出

```
77.53982
39.26991
```