

2001 年中山大学第四届大学生程序设计竞赛

暨第 26 届 ACM/ICPC 中山大学组队选拔赛

中山大学教务处主办

中山大学信息科学与技术学院计算机科学系实施
中山大学信息学研究实验室命题



中山大学，广州/中山大学珠海校区，珠海

第二轮竞赛

2001 年 5 月 13 日

这份题目应当有 8 页，共 5 题。

如果发现缺页请立即联系竞赛工作人员。



注意事项

欢迎参加 2001 年中山大学第四届大学生程序设计竞赛暨第 26 届 ACM/ICPC 中山大学组队选拔赛。为了能够更好地享受竞赛的乐趣，请注意以下事项：

1. 竞赛时间 4 小时，共 5 道题。
2. 遵守竞赛的规则。请不要携带任何书籍资料、计算器和计算器可读的媒体（如磁盘）进场。竞赛结束后，请遵从竞赛工作人员的指示离场。
3. 你可以按照自己喜欢的顺序解题，不必遵守题目实际出现的顺序。你也可以按照自己喜欢的风格编写程序。在竞赛中不需要写任何文档。
4. 请将源程序编译成可执行文件，注意可执行文件名应符合题目的要求。不要使用非标准的库（另外，使用 Pascal 的选手请注意不要使用 CRT 单元，使用 C/C++ 的选手请注意不要使用 conio.h）。如果你在编译时有任何困难，请立即联系竞赛工作人员。竞赛结束后，请不要对源程序作任何改动。
5. 请按照题目的要求输入输出。
 - 不要自行更改输入输出的方式。对采用文件输入输出的题目，请从指定的文本文件中读取数据，输出到指定的文本文件，程序在执行过程中不需要与人进行交互。文件名中不要包含驱动器符号和绝对路径，你可以认为输入输出文件与可执行文件在同一个目录下。在竞赛期间，请使用文本编辑器自行创建输入文件。
 - 在数据输入时，请不要更改数据输入的格式、次序、换行要求和数据的分隔字符，也不要输出任何的提示信息，即使这样的更改可能有利简化程序。除了题目声明之外，请不要对数据的范围作任何的假设，你可以认为所有的输入都符合题目规范，不需要编程处理不符合题目规范的输入。
 - 请严格按照题目要求的格式输出。程序的输出的格式应当与样例输出的格式完全一致。包括对单词拼写（包括错误拼写）、字母大小写、标点、空格和换行的要求。请不要使用制表字符（Tab）。除了题目要求之外，不要输出多余空行，或在行末输出多余的空格，也不要连续输出多个空格。
6. 在竞赛期间，每隔一定时间请自行将源程序存盘备份，避免机器故障带来麻烦。
7. 竞赛评委的决定是最终的。
8. 如果你发现机器有问题或对竞赛有任何疑问，请立即联系竞赛工作人员。

第一题 超级分子 (40 分)

可执行文件: molecule.exe

输入方式: 键盘输入

输出方式: 屏幕输出

这个问题是在与人造燃料研制相关的分子工程学当中抽象出来的: 用四个等长的分子链合成一个超级分子。在这里我们使用一种简化的二维模型: 超级分子是由四条两两平行、相互扣在一起的分子链形成的。“扣在一起”, 即两条分子链共享一个分子。

举例来说, 假设我们有四个长度都是 12 的分子链:

```
O I M D I H E I A F N L
C H J D B J M H P J K D
L C B J O J G I E K B O
K A I N L H L O L B E J
```

它们可以合成下面的超级分子:

```

O           L           O           C
I           C           I           H
M           B           M           J
C H J D B J M H P J K D           D           D
I           O           L C B J O J G I E K B O
H           J           H           J
E           G           E           M
I           I           K A I N L H L O L B E J
A           E           A           P
F           K           F           J
K A I N L H L O L B E J           N           K
L           O           L           D
    
```

- 或 -

在这个问题里, 我们对超级分子的结构有若干限制:

1. 四条分子链中任何一条都可以放在超级分子的水平或垂直“槽”中, 如上图所示。
2. 如果一条分子链放在超级分子的两条水平槽之一, 它必须保持原本从左到右的顺序, 也就是说, 它不可以反转。
3. 如果一条分子链放在超级分子的两条垂直槽之一, 它从上到下的顺序必须和原本从左到右的顺序相同, 换句话说, 它也不可以反转。
4. 超级分子中心围起来的矩形区域的面积必须尽可能大, 而且不能是零。面积尽量大的限制是从燃料挥发性的准则而得来的。面积有非零的限制, 是因为不论水平还是垂直分子链彼此接触都会引起某种副作用。
5. 所有分子链两端都应当延伸出超级分子中心的矩形区域至少一个分子的长度。换言之, 没有分子链的第一个或最后一个分子能成为超级分子中心的矩形边界的一部分。

输入

输入有四行, 每行 12 个字符, 分别代表四个长度为 12 的分子链。每种分子用一个大写字母表示。由于分子数目的限制, 我们将只使用 A 到 P 这 16 个大写字母。分子链的第一个分子就是每行的第一个字符。

输出



请将四条分子链合成的超级分子中心的矩形区域的面积按以下样例的格式输出。这个数字应当是给定的四条分子链所能合成的所有超级分子中心的矩形区域面积的最大值。

如果给定的四条分子链无法合成超级分子，请输出 0。

输入样例

输出样例

CDBADCBBEFEF DACCBADAFEAB EFBDCAADBDCD ABCDABCDABCD	48
ABABABABABAB CDCDCDCDCDCD EEEEEEEEEEEEEE FFFFFFFFFFFFFFF	0

第二题 大众披萨？（40 分）

可执行文件：pizza.exe

输入文件：pizza.in

输出文件：pizza.out

你准备为你和朋友们订一个披萨，他们告诉你每个人希望披萨里有和没有什么材料。当然，他们也明白到只有一个披萨，所以没有人期望他所有的要求都得到满足。你能订一个披萨满足他们每个人至少一项要求吗？

你订披萨的那间店子提供如下的披萨口味，你可以在一个披萨中要或不要当中任何一项：

代码	口味
A	凤尾鱼 (Anchovies)
B	黑橄榄 (Black Olives)
C	加拿大熏肉 (Canadian Bacon)
D	方丁大蒜 (Diced Garlic)
E	浓奶酪 (Extra Cheese)
F	鲜花椰菜 (Fresh Broccoli)
G	青胡椒 (Green Peppers)
H	火腿 (Ham)
I	意大利香肠 (Italian Sausage)
J	加拉佩诺胡椒 (Jalapeno Peppers)
K	波兰熏肠 (Kielbasa)
L	瘦牛肉 (Lean Ground Beef)
M	蘑菇 (Mushrooms)
N	脱脂羊乳酪 (Nonfat Feta Cheese)
O	洋葱 (Onions)
P	胡椒 (Pepperoni)

你的朋友每人给你一行文字描述他们的披萨口味。例如，

+O-H+P;

意味着某位朋友将接受一个不含洋葱，或不含火腿，或带有胡椒的披萨。而

-E-I-D+A+J;

表示某位朋友将接受一个不含浓奶酪或意大利香肠或方丁大蒜的，或带有凤尾鱼或加拉佩诺胡椒的披萨。

输入

输入文件 `pizza.in` 有若干行，描述了对一个披萨的约束。一个披萨约束由 1 到 12 个口味约束组成，一个口味约束占一行，表示一位朋友的口味。最后一行仅含一个句号“.”，表示输入的开始。

一个口味约束是一组由分号“;”结束的口味要求序列。

一个口味要求由一个符号 (+/-) 接着一个从 A 到 P 的大写字母组成。

输出

请将任一种符合要求的披萨的描述输出到文件 `pizza.out`。一个披萨描述由一个长度为 10 的字符串“Toppings: ”开始（注意有一个空格），接着是一组按字母序排列的字母，列出披萨的口味。例如，一个带有洋葱、凤尾鱼、鲜花椰菜和加拿大熏肉的披萨描述为：
Toppings: ACFO

如果没有口味的组合可以满足每人至少一项请求，你的程序应输出字符串

“No pizza can satisfy these requests.”

输入样例

输出样例

<pre>+A+B+C+D-E-F-G-H; -A-B+C+D-E-F+G+H; -A+B-C+D-E+F-G+H; .</pre>	<pre>Toppings:</pre>
<pre>+A+B+C+D; +E+F+F+H; +A+B-G; +O+J-F; +H+I+C; +P; +O+M+L; +M-L+P; .</pre>	<pre>Toppings: CELP</pre>
<pre>+A+B+C+D; +E+F+F+H; +A+B-G; +P-O; +O+J-F; +H+I+C; +P; +O;</pre>	<pre>No pizza can satisfy these requests.</pre>



+O+M+L; -O-P; +M-L+P; .	
----------------------------------	--

第三题 UNIX 的插头 (40 分)

可执行文件: plug.exe

输入文件: plug.in

输出文件: plug.out

你负责为联合国互联网执行组织 (UNIX) 的周年会议布置会议室。该组织的主旨，是使互联网上的信息和思想的自由交流变得尽可能笨重和官僚 (^_^)。

因为会议室被设计为招待来自世界各地的记者，它装备了一些电气插座以匹配各个国家使用的电器的不同插头形状和电压。当然，这些插座的规格，是以这个会议室建成时世界各国的规格作为标准的。不幸的是，由于这座会议室年代久远，它建成时，记者还只是使用很少量的电子和电气设备，所以它对每种类型只配备了一个插座。今天，像其它人一样，记者需要很多设备才能工作：掌上型计算器，手提电话，录音机，寻呼机，电咖啡壶，微波炉，电吹风，电熨斗，电动牙刷，等等。自然，这些设备当中有许多可以靠电池驱动的，但是看起来这个会议将会冗长乏味，所以还是希望这些设备尽可能从插座供电。

在会议开始前，你将记者所有可能使用的设备集中起来，尝试将它们接到插座上。你注意到有些设备的插头没有相应的插座。你怀疑这些设备来自一些在这座会议室建成时还不存在的国家。对某些插座，也许有许多设备都使用对应的插头；而对另一些插座，也许没有任何设备使用对应的插头。

为了解决这个问题，你走到附近一家五金电气商店。这家商店出售一些转接头，可以将一种插头的转换成另一种插头。一个转接头还可以接到另一个转接头上。并非所有插座和插头的组合都有相应的转接头，但是对每种出售的转接头，商店都有足够的存货。

输入

输入文件 plug.in 有若干行。第一行是一个整数 n ($1 \leq n \leq 100$)，表示会议室的插座数。接下来的 n 行列出了能在会议室找到的插座，每种插座的名称是由字母和数字组成的字符串，长度不超过 24 个字符。接着一行是一个整数 m ($1 \leq m \leq 100$)，表示你要连接的插头数。再接下来的 m 行列出了设备的名称和它使用的插头（与相应插座的名称相同）。两者用一个空格分隔。每种设备的名称也是由字母和数字组成的字符串，长度不超过 24 个字符。没有两个设备的名称完全相同。接着一行是一个整数 k ($1 \leq k \leq 100$)，表示可用的转接头种类。接下来的 k 行每行描述了一种转接头，首先给出转接头提供的插座，接着一个空格，接下来是插头的类型。

输出

请将最少的无法连接的设备数输出到文件 plug.out。

输入样例

输出样例

4	1
A	
B	
C	
D	
5	
laptop B	
phone C	
pager B	
clock B	
comb X	
3	
B X	
X A	
X D	

第四题 街道的方向 (40 分)

可执行文件: `street.exe`

输入文件: `street.in`

输出文件: `street.out`

汽车碰撞监控 (ACM) 的一份报告指出, 大部分严重车祸是在双行道上发生的。为了减少车祸的伤亡, 市长决定将尽可能多的街道改成单行道。你的任务是完成这项工作, 同时满足司机在任何一个路口, 经过某条路线能到达其它任何一个路口。

你将收到这个城市的街道资料 (所有的街道都是双行道)。每条街道连接两个路口, 并且不会穿越其它的路口。最多有四条街道汇集在同一个路口, 而且任意两个路口最多有一条街道连接。可能有某个路口只是某条街道的一端 (即死胡同), 也有可能改道完成后, 仍有两个路口之间的路径所经过的街道全是双行道。

输入

输入文件 `street.in` 有若干行, 第一行是两个用空格分隔的整数 n 和 m , 分别表示路口数 ($1 \leq n \leq 1000$) 和街道数。路口从 1 到 n 进行编号。接下来的 m 行, 每行包含两个用空格分隔的整数 i 和 j , 表示连接路口 i 和 j 的街道。每条街道只出现一次, 如果街道 $i j$ 出现过, 则街道 $j i$ 就不会再出现。

输出

请将改道的方案输出到文件 `street.out`。对每条从路口 i 到 j 的单行道, 输出一行 $i j$ 。对每条不能改成单行道的街道 $i j$, 分开两行输出 $i j$ 和 $j i$ 。街道的两个路口用一个空格分隔。你可以用任何顺序输出街道。

注意: 也许有多个符合题目要求的方案, 你可以输出任何一个。

输入样例

输出样例

7 10 1 2 1 3 2 4 3 4 4 5 4 6 5 7 6 7 2 5 3 6	1 2 2 4 3 1 3 6 4 3 5 2 5 4 6 4 6 7 7 5
7 9 1 2 1 3 1 4 2 4 3 4 4 5 5 6 5 7 7 6	1 2 2 4 3 1 4 1 4 3 4 5 5 4 5 6 6 7 7 5

第五题 逻辑岛（40 分）

可执行文件: island.exe

输入文件: island.in

输出文件: island.out

逻辑岛上有三种居民：天使 (divine) 永远说真话，魔鬼 (evil) 永远说谎话，人类 (human) 白天说真话，晚上说谎话。岛上每个居民都知道其它居民的身份。

一个社会科学家来到逻辑岛。因为他无法从居民的相貌区分她们的身份，他希望你能设计一个通讯分析器，从居民之间的谈话中推导一些事实。他感兴趣的事实是当时是白天还是晚上，以及说话者是何种居民。

输入

输入文件 island.in 有若干行，第一行是一个整数 n，表示对话包含的句子数。接下来的 n 行，每行包含一句居民的话。每句话由说话者的名称 (A, B, C, D, E 五个大写字母之一) 开始，接着一个冒号 “:” 和一个空格，接下来是下面三种句子之一：

- I am [not] (divine | human | evil | lying).
- X is [not] (divine | human | evil | lying).
- It is (day | night).

方括号 [] 表示括号中的单词可以出现也可以不出现，圆括号 () 表示由 | 分隔的若干项之一必须出现。X 是 A, B, C, D, E 五个名称之一。句子当中不会出现两个连续的空格。

对话中最多有 50 个句子。

输出

请将能够推导的事实输出到文件 island.out。如果根据规则不可能发生这样的对话，请输出 “This is impossible.”。如果无法推导出任何事实，请输出 “No facts are deducible.”。否则请按照下面的格式输出所有能够推导的事实：

- X is (divine | human | evil) .
- It is (day | night) .

x 应当替换为相应说话者的名称。请先按字母序输出关于居民身份的事实，然后是关于白天黑夜的事实。

输出时，事实的前后应当都没有空格，事实当中也不会出现两个连续的空格。

输入样例

输出样例

1 A: I am divine.	No facts are deducible.
1 A: I am lying.	This is impossible.
1 A: I am evil.	A is human. It is night.
3 A: B is human. B: A is evil. A: B is evil.	A is evil. B is divine.

提示

为了使你更好地理解推理的过程，我们来解释一下第三个样例的推理。A 说 “I am evil.” 从中能推导出什么？显然，A 不可能是天使，否则她就在说谎；类似的，A 也不可能是魔鬼，否则她就在说真话。所以，A 一定是人类，而且，由于她在说谎，当时一定是晚上。所以正确的输出如样例所示。

在第四个样例里，显然 A 在说谎，因为她的话前后矛盾。所以 B 既不是人类也不是魔鬼，因此 B 一定是天使。B 永远说真话，所以 A 一定是魔鬼。哈哈！