

黄冈中学信息奥赛 CSP-S 训练题

测试时间 14:30—18:30

(请仔细阅读本页面内容)

一. 题目概况

中文题目名称	玩具谜题	教官整队	秘密信息	括号序列
英文题目与子目录名	toy	queue	hidden	brackets
可执行文件名	toy	queue	hidden	brackets
输入文件名	toy.in	queue.in	hidden.in	brackets.in
输出文件名	toy.out	queue.out	hidden.out	brackets.out
每个测试点时限	1s	1s	1s	1s
测试点数目	20	10	10	10
每个测试点分值	5	10	10	10
附加样例文件	有	有	有	有
结果比较方式	全文比较（过滤行末空格及文末换行）			
题目类型	传统	传统	传统	传统
运行内存上限	256M	256M	256M	256M

二. 提交源程序文件名

对于 C++ 文件	ticket.cpp	queue.cpp	hidden.cpp	brackets.cpp
对于 C 文件	ticket.c	queue.c	hidden.c	brackets.c

三. 编译命令 ((不包含任何优化开关))

对于 C++ 文件	g++.exe %s.cpp -o %s.exe -lm
对于 C 文件	gcc.exe %s.c -o %s.exe -lm

注意事项:

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中的函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 评测时不推荐使用万能头文件 `#include <bits/stdc++.h>`，是否支持 c++11 详见题目说明。
4. 提交的文件目录结构如下图所示，HB-0001 为考生准考证号，calm、steady、strength 为题目规定的英文名称。目录中只包含源程序，不能包含其他任何文件。如有特殊说明，请严格按照格式要求提交源代码，否则可能导致评测失败。

```

HB-0001 ----- 准考证号建立文件夹
  calm ----- calm.cpp
  steady ----- steady.cpp
  strength ----- strength.cp
  
```

玩具谜题(toy)

【题目描述】

小凯有一套可爱的玩具小人，它们各有不同的职业。

有一天，这些玩具小人把小凯的眼镜藏了起来。小凯发现玩具小人们围成了一个圈，它们有的面朝圈内，有的面朝圈外。如下图所示：

这时 singer 告诉小凯一个谜题：“眼镜藏在我左数第 3 个玩具小人的右数第 1 个玩具小人的左数第 2 个玩具小人那里。”



小凯发现，这个谜题中玩具小人的朝向非常关键，因为朝内和朝外的玩具小人的左右方向是相反的：面朝圈内的玩具小人，它的左边是顺时针方向，右边是逆时针方向；而面向圈外的玩具小人，它的左边是逆时针方向，右边是顺时针方向。

小凯一边艰难地辨认着玩具小人，一边数着：

singer 朝内，左数第 3 个是 archer。

archer 朝外，右数第 1 个是 thinker。

thinker 朝外，左数第 2 个是 write。

所以眼镜藏在 writer 这里！

虽然成功找回了眼镜，但小凯并没有放心。如果下次有更多的玩具小人藏他的眼镜，或是谜题的长度更长，他可能就无法找到眼镜了。所以小凯希望你写程序帮他解决类似的谜题。这样的谜题具体可以描述为：

有 n 个玩具小人围成一圈，已知它们的职业和朝向。现在第 1 个玩具小人告诉小凯一个包含 m 条指令的谜题，其中第 i 条指令形如“左数/右数第 s_i 个玩具小人”。你需要输出依次数完这些指令后，到达的玩具小人的职业。

【输入格式】

从文件 toy.in 中读入数据。

接下来 n 行，每行包含一个整数和一个字符串，以逆时针为顺序给出每个玩具小人的朝向和职业。其中 **0** 表示朝向圈内，**1** 表示朝向圈外。保证不会出现其他的数。字符串长度不超过 **10** 且仅由字母构成，字符串不为空，并且字符串两两不同。整数和字符串之间用一个空格隔开。

接下来 m 行，其中第 i 行包含两个整数 a_i, s_i ，表示第 i 条指令。若 $a_i=0$ ，表示向左数 s_i 个人；若 $a_i=1$ ，表示向右数 s_i 个人。保证 a_i 不会出现其他的数， $1 \leq s_i < n$ 。

【输出格式】

输出到文件 `toy.out` 中。

输出一个字符串，表示从第一个读入的小人开始，依次数完 m 条指令后到达的小人的职业。

【样例输入 1】

```
7 3
0 singer
0 reader
0 mengbier
1 thinker
1 archer
0 writer
1 magician
0 3
1 1
0 2
```

【样例输出 1】

```
writer
```

【样例输入 2】

```
10 10
1 c
0 r
0 P
1 d
1 e
1 m
1 t
1 y
1 u
0 v
1 7
1 1
1 4
```

0 5
0 3
0 1
1 6
1 2
0 8
0 4

【样例输出 2】

y

【数据范围与约定】

对于 80%的数据， $n \leq 20, m \leq 1000$ 。

对于 100%的数据， $n, m \leq 100000$ 。

每个测试点的数据规模及特点如下表：

测试点	n	m	全朝内	全左数	Si= 1	职业长度为1
1	20	1000	√	√	√	√
2			x			
3			√	x		
4			x			
5			√	√	x	
6			x			
7			√	x		
8			x			
9			√	√	√	
10			x			
11			√	x		
12			x			
13			√	√	x	
14			x			
15			√	x		
16			x			
17	100000	100000	√	√		x
18			x			
19			√	x		
20			x			

教官整队(queue)

【题目描述】

小凯如愿以偿的考上了黄冈中学,新生报到的第二天就迎来了军训。这一天,他们班的同学正在进行队列练习。教官希望他们站成一排,然而很糟糕,本该站成一条直线的队伍却像一条长蛇一样弯弯曲曲。如果把训练场看成一个平面直角坐标系,第 i 名同学所在位置的横坐标是 i ,那么所有同学的纵坐标是 0 (或者任意一个相等的常量),这样就排成了一条直线。当然,由于同学们排的歪歪扭扭,所以第 i 位同学的横坐标依然是 i ,而纵坐标却成了 Y_i (为了简化问题,我们假设所有的坐标都是整数)。对此,教官当然十分生气,因此他决定下达命令调整队伍,使得所有人能够站成一条直线(也就是让所有的 Y_i 相同)。

教官的命令总共有三种:

- 1、除了某一个同学外,其余所有同学向前一步走(向前走一步可理解为 Y_i 的值加 1,下同);
- 2、除了某一个同学外,其余所有同学向前两步走;
- 3、除了某一个同学外,其余所有同学向前五步走。

教官希望他能以最少的命令次数使得所有同学站成一条直线,但是他不会算,于是就让小凯帮他计算。小凯虽然很聪明,但由于班上人数众多,他一下子也解决不了这个问题,所以只能求助于黄冈中学信息奥赛小组,要你帮他编个程序解决这个问题。

【输入格式】

从文件 `queue.in` 中读入数据。

第一行有一个整数 n ,表示班上共有 n 位同学。

第二行有 n 个整数,第 i 个整数 Y_i 表示第 i 位同学初始位置的纵坐标。

【输出格式】

输出到文件 `queue.out` 中。

一个整数,表示教官最少下达的命令次数。

【样例输入】

```
4
1 1 2 6
```

【样例输出】

```
2
```

【样例说明】

一种方案是: $1\ 1\ 2\ 6 \rightarrow 2\ 2\ 2\ 7 \rightarrow 7\ 7\ 7\ 7$

【数据范围与约定】

对于 60% 的数据, $n \leq 1000$, $0 \leq Y_i \leq 1000$;

对于 100% 的数据, $1 \leq n \leq 100000$, $0 \leq Y_i \leq 10^9$ 。

秘密信息 (hidden)

【题目描述】

小凯想用以下的方法加密一条信息（这是她从密码学书上自学来的）：

假定这条信息可以用一个字符串 S 表示，其中 $S=BCAAD.$ （其中 ‘.’ 代表字符串结尾）。小凯首先把 S 的所有循环同构串写下来（所谓循环同构即是不不断地把字符串开头的字符移动到尾端）：

BCAAD.
CAAD. B
AAD. BC
AD. BCA
D. BCAA
. BCAAD

接下来她会把这些字符串按字典序排序：

. BCAAD
AAD. BC
AD. BCA
BCAAD.
CAAD. B
D. BCAA

接下来她会把这些字符串的最后一位按照顺序写下来得到加密串 $T=DCA. BA$

现在小凯发现在密码学的书上写着一条来历不明的加密串 T ，她迫切地知道，如果信息是按照这种方式加密的，那么原串是什么？

【输入格式】

从文件 `hidden.in` 中读入数据。

输入一行，一个由大写字母构成长度为 N 的字符串（字符串有且只有一个 ‘.’’）

【输出格式】

输出到文件 `hidden.out` 中。

输出一行，输出原串，保证数据有唯一解。

【样例输入 1】

PH. EL

【样例输出 1】

HELP.

【样例输入 2】

BBA. AA

【样例输出 2】

ABAAB.

【数据范围与约定】

对于 20%的数据， $2 \leq N \leq 10$ 。

对于 50%的数据， $2 \leq N \leq 27$ ，字符串内部的字母互不相同。

对于 100%的数据， $2 \leq N \leq 30000$ 。

括号序列(brackets)

【题目描述】

课堂上,小凯刚刚学习了关于括号序列的知识。括号序列是一个只由左括号“(”和右括号“)”构成的序列;进一步的,一个合法的括号序列是指左括号和右括号能够一一匹配的序列。

如果用规范的语言说明,一个合法的括号序列可以有以下三种形式:

- 1、 $S = ""$ (空串), S 是一个合法的括号序列;
- 2、 $S = XY$, 其中 X, Y 均为合法的括号序列, 则 S 也是一个合法的括号序列;
- 3、 $S = (X)$, 其中 X 为合法的括号序列, 则 S 也是一个合法的括号序列。

这时老师在黑板上写出了一个了括号序列:“())) ()”。小凯一眼就看出这个序列并不是合法的括号序列。这时老师提出了一个这样的问题:能否在序列中找出连续的一段,把这一段里面的左括号变成右括号,右括号变成左括号,变换之后整个序列可以变成合法的呢?

小凯想到,可以把[3..5]进行调换,这样序列就会变为() () ,是一个合法的序列。很明显,不止有一种方法可以使整个序列变合法。

这时,老师又在黑板上写出了长度为 N 的括号序列。小凯想,能否对这个序列进行最多一次变换,使它变合法呢?

【输入格式】

从文件 brackets.in 中读入数据。

第一行一个整数 T , 代表数据的组数; 接下来 T 行, 每一行一组数据。

每组数据一行, 代表给出的括号序列。

【输出格式】

输出到文件 brackets.out 中。

输出共 T 行, 对于每组数据, 输出 “possible”(可以变换) 或 “impossible”(不可变换)。(注意不含引号)

【样例输入】

```
3
( )))
))) (
()
```

【样例输出】

```
possible
impossible
possible
```

【数据范围与约定】

对于 50% 的数据, $T \leq 5$, $N \leq 20$

对于 100% 的数据, $T \leq 10$, $N \leq 5000$