

hello

起床是一只仓鼠，自从听说ACM比赛之后，便想要参加比赛，奈何每日赖床不起，没时间学习。某日，起床在梦中梦到自己逛书店，看到了《C语言程序设计》，其中第一个程序便是输出"hello world!"。"这也太简单了"，起床自信地说："我要写一个比这更难的程序！"。起床想对于任意给出的字符串S，都能按以下格式输出："hello "+S+"!"。例如：S为"ACM"，则输出"hello ACM!"。

input

输入只有一行，包含一个字符串S，S的长度不超过10。字符串只包含大小写字母，数字及下划线。

output

输出一行，格式为"hello "+S+"!"。

样例输入1

复制

```
world
```

样例输出1

复制

```
hello world!
```

样例输入2

复制

```
ACMer
```

样例输出2

复制

```
hello ACMer!
```

cola

起床喜欢喝可乐，它有一个瓶子，瓶子里有体积为 n 的可乐。另外有 m 个杯子，每个杯子的容量为 $p(i)$, ($1 \leq i \leq m$)，现在要把瓶子里的可乐倒入杯子里，起床想知道能否把瓶子里的可乐倒光，如果可以，最少需要用多少个杯子。

input

第一行有两个整数 n, m ，表示可乐的体积和杯子的个数($0 \leq n \leq 1000, 0 \leq m \leq 10$)。接下来一行有 m 个整数，第 i 个整数代表 $p(i)$ ，($1 \leq p(i) \leq 100$)。

output

如果可以把瓶子里的可乐倒光，输出一行包含一个整数，表示最少需要杯子的数量，否则输出一行"NO"。

样例输入1

[复制](#)

```
10 5
3 5 2 1 3
```

样例输出1

[复制](#)

```
3
```

样例输入2

[复制](#)

```
10 2
3 5
```

样例输出2

[复制](#)

```
NO
```

fengbao

FengBaoYingXiong是由BaoXue娱乐公司开发的一款在线多人竞技MOBA类游戏。起床作为BaoXue全家桶的受害者，自然也中了这款游戏的毒。然而无奈的是，玩这款游戏的仓鼠似乎并不怎么多，起床有时候匹配好久也进入不了游戏。为了帮助这款游戏火起来，经常有仓鼠在论坛上发帖说:"FengBaoYaoHuo"。

于是起床从论坛上收集了一些帖子，每个帖子由两个字符串组成，分别是仓鼠的昵称，和它的发言。起床想要统计有多少只不同的仓鼠发言了"FengBaoYaoHuo"，并把结果保存在整数S中。然而，有一只来自东方的神秘仓鼠叫做"HuangXuDong"，如果HuangXuDong也说: "FengBaoYaoHuo"，或者S为0，那么就输出"FengBaoLiangLe"，否则就输出"FengBaoYaoHuo = "+S。

另外仓鼠们在论坛发言时并不注意大小写格式，那么看仓鼠的发言时应当大小写不敏感。也就是说，无论发言为"fengbaoyaohuo"或"FENGBAOYAOHUO"或"fengBAOYAOhuo"等等，都视作发言为"FengBaoYaoHuo"。

input

输入第一行有一个整数n, ($1 \leq n \leq 100000$), 表示起床收集的帖子的个数。接下来n行，每行两个字符串，中间用一个空格分隔，代表仓鼠昵称和它的发言。保证每个字符串只包含大小写字母和数字及下划线，且长度不超过20。

ouput

输出一行，如果存在昵称为HuangXuDong，且发言为"FengBaoYaoHuo"，或者S为0，那么就输出"FengBaoLiangLe"，否则就输出"FengBaoYaoHuo = "+S。

样例输入1

[复制](#)

```
8
dog hello
cat fengbaoyaohuo
dog fengbaoyaohuo
frog fengbaoliangle
HuangXuDong fengbaoliangle
CAT fengbaoyaohuo
dog FengBaoYaoHuo
frog FENGBAOYAOHUO
```

样例输出1

[复制](#)

FengBaoYaoHuo = 4

样例输入2

复制

2
dog fengbaoyaohuo
HuangXuDong FengBaoYaoHuo

样例输出2

复制

FengBaoLiangLe

chocolate

水饺是一只贪吃的仓鼠，尤其喜欢吃巧克力。而且每吃一个巧克力，它就会获得一定的喜悦度（由巧克力决定），并且连续吃巧克力时，喜悦度会以恐怖的速度增长。现在有 N 个巧克力，每个巧克力都有一个喜悦度 $a[i]$, ($1 \leq i \leq N$)，当水饺吃掉第 i 个巧克力后，所有其他巧克力的喜悦度都会增长 $a[i]$ 。水饺想要在吃完所有 N 个巧克力后，获得最大的喜悦度，问水饺应该怎样吃巧克力，并按顺序输出每次吃巧克力会得到多少的喜悦度。

input

输入第一行有一个整数 N ，($1 \leq N \leq 50$)表示巧克力的个数。接下来 n 行，每行一个数字 $a[i]$ ，($0 \leq a[i] \leq 10$)表示第 i 个巧克力的喜悦度。

output

输出 N 行，每行一个整数，第 i 行的数字代表第 i 次吃掉的巧克力有多少喜悦度。

样例输入

复制

```
3
2
1
2
```

样例输出

复制

```
2
4
7
```

fibonacci

水饺碰到不会的问题经常去问起床，起床也很乐意解答。今天，水饺又碰到了个难题。“你知道斐波那契数列吗？”，水饺问起床。“当然！”，起床说：“斐波那契数列指的是这样一个数列：1、1、2、3、5、8、13、21、34、.....在数学上，斐波纳契数列以如下被以递推的方法定义： $F(1)=1$ ， $F(2)=1$ ， $F(3)=2$ ， $F(n)=F(n-1)+F(n-2)$ ($n \geq 4$ ， $n \in \mathbb{N}^*$)”。水饺高兴地说：“这个问题就是查询斐波那契数列第 n 项的值，你帮我看看吧！”。起床拿到问题一看，却也被难到了，只好向你寻求帮助，你能帮它解决这道题目吗？

input

第一行一个整数 T ，($1 \leq T \leq 100000$)，代表查询的数量。接下来 T 行，每行一个整数 n ，($1 \leq n \leq 1000000000$)，代表查询斐波那契数列的第 n 项。

output

对于每个查询输出一个整数，每个查询占一行，因为答案可能过大，只需输出 $F(n) \% 1000000007$ 的结果。

样例输入

[复制](#)

```
6
1
2
3
4
5
1000
```

样例输出

[复制](#)

```
1
1
2
3
5
517691607
```

follow

经过一段时间刻苦的学习，起床去参加了一次ACM赛制的程序设计竞赛。然而一下子面对这么多的题目，起床并不知道该先解决哪个比较好。有一个传统的解决方法叫做“跟榜”，就是查看实时的榜单，从中选出通过数最多且自己还没通过的题目去做。然而跟榜也容易出事故，假如有大佬们开始都做的是比较难的题，后面的仓鼠都跟着去做，导致简单的题目却无人问津。

不过起床也管不了那么多，它也打算使用“跟榜”这个方法去选择接下来做的题。假如参加这场比赛的共有 n 只仓鼠，（每只仓鼠为一队），有 m 道题目（题目序号为 $1-m$ ），起床目前排在第 p 位，然后给出目前的简略榜单。榜单共 n 行，每行有 m 个数字（0或1），第 i 行（ $1 \leq i \leq n$ ）代表排名第 i 的仓鼠通过题目的情况，第 p 行就代表起床通过题目的情况。每行的第 j 个（ $1 \leq j \leq m$ ）数字代表这只仓鼠是否通过了第 j 道题，1代表通过了，0代表未通过。

起床想找到通过数最多，且自己还未通过的题目，如果这样的题目有多个，选择序号最小的那个。保证起床没有通过所有题目。

input

输入第一行为3个整数 n, m, p ，（ $2 \leq n \leq 100, 2 \leq m \leq 15, 1 \leq p \leq n$ ）代表参加比赛的仓鼠个数，题目个数，起床的排名。接下来 n 行，每行 m 个整数（1或0），第 i 行第 j 个数代表排名第 i 的仓鼠是否通过了第 j 道题目（1为通过，0为未通过）。

output

输出一行包含一个整数，代表起床应该选择的题目序号。

样例输入1

[复制](#)

```
5 5 5
1 1 1 0 1
0 1 1 1 0
1 0 1 1 0
0 1 1 1 0
0 0 1 0 0
```

样例输出1

[复制](#)

```
2
```

样例输入2

[复制](#)

```
5 3 1
1 1 0
1 1 0
1 0 0
1 0 0
0 0 0
```

样例输出2

复制

```
3
```


acm

比赛结束后，工作仓鼠们正准备发布最后的终榜，可是比赛服务器突然崩溃了，当机器重启后，它们惊恐的发现榜单信息损坏了。幸好比赛过程中的所有的提交记录还在，这次比赛的工作仓鼠经验还不足，希望能找仓鼠帮忙恢复榜单，如果谁能帮忙利用好提交记录恢复榜单，便给它颁发一个特等奖。起床听到后立马自告奋勇，写一个程序解决这个问题。

输入一开始会给出两个数字N,M,K，表示参加比赛选手数量，题目数量，和总共提交记录的数量。然后给出参加选手的名单，还有提交记录。每个名单是一个字符串，独占一行，名单中没有相同的两个名字。每个提交记录由四个空格分隔的字符串构成，也独占一行，分别表示选手名字，提交题号，提交时间，提交结果四个信息，所有提交都已按照提交时间递增顺序排好。

选手名字表示这次提交是由哪位选手提交的，且保证是在选手名单中出现过的。

提交题号为一个大写英文字母，A代表第一题，B是第二题，C是第三题...，以此类推，且不会超过题数M的限制，M最大为26。

提交时间格式为 A:B，表示这次提交的时间是在比赛开始后第A小时第B分钟。A为不大于10的非负整数，B为不大于59的非负整数，A,B在不够两位数的情况下会补前导0至两位，如"04:07"，表示时间四小时七分钟。

提交结果有以下几种：AC(通过)，WA(答案错误)，TLE(运行超时)，MLE(内存超限)，RE(运行时出错)，CE(编译错误)。

每个选手有四种数据，分别是通过题目数量，罚时，和每道题的总共提交次数和错误提交次数。这次比赛为ACM赛制，选手们先按照通过题目数递减，再按照罚时时间递增排名，如果通过题数和罚时都相同的选手，再按照字典序递增顺序排名。

一开始所有选手的所有数据都为0。对于每个提交，如果对于提交的选手来说，提交的题目已经通过了，则这次提交不会产生任何影响。如果是没有通过的题目，首先这道题的总共提交次数+1，再对于不同的提交结果，

[1]:结果为AC，那么这道题算作通过，选手的通过题目数+1，罚时+=(这道题目的错误提交次数*20+比赛开始经过的时间)，比赛开始经过时间=60*A+B，(A,B的含义见提交时间格式中的描述)，

[2]:结果为CE，没有影响，

[3]:结果为其他，则这道题的错误提交次数+1。

在处理完所有提交产生排名后，按照排名输出榜单，每人占一行。一行中首先是一个字符串，表示选手名字，接下来两个整数，表示通过题目数量和罚时，再接下来K个整数，第i个整数表示第i题的提交情况，(如果没有提交过或者提交过且通过，数字为 这道题的总共提交次数，否则数字为 0-这道题的总共提交次数)。

实际的ACM比赛情况可能和这道题目略有不同，编写程序时应按照题目所说为准。

input

第一行三个整数N,M,K,($1 \leq N \leq 300$, $1 \leq M \leq 26$, $0 \leq K \leq 30000$)，含义如题目所说。接下来N行，每行一个字符串，按照字典序给出选手名单，字符串长度不超过20，且只包含大小写英文字母。再接下来K行，每行四个空格分隔的字符串，分别表示选手名字，提交题号，提交时间，提交结果四个信息，格式见题目及样例。

output

输出N行，表示排名后的榜单。每行首先是一个字符串，表示选手名字，接下来两个整数，表示通过题目数量和罚时，再接下来K个整数，第i个整数表示第i题的提交情况，具体格式见题目及样例。

样例输入

[复制](#)

```
5 3 10
cat
dog
frog
qichuang
shuijiao
cat A 00:05 AC
dog A 00:06 AC
qichuang A 00:10 CE
shuijiao A 00:15 AC
cat B 00:20 AC
dog B 00:21 WA
qichuang A 00:25 WA
shuijiao B 00:30 TLE
dog B 00:35 AC
qichuang A 00:40 AC
```

样例输出

[复制](#)

```
cat 2 25 1 1 0
dog 2 61 1 2 0
shuijiao 1 15 1 -1 0
qichuang 1 60 3 0 0
frog 0 0 0 0 0
```

tree

水饺和起床相约一起去野外探险，它们来到了一颗巨大的果树旁。果树可以看作是由 N 个节点构成的一棵树，节点序号为 $(1-N)$ ，每个节点上都有一个果子，大小为 $a[i]$ 。除了根节点外，每个节点都有一个父节点，并且如果 A 是 B 的父节点，那么 B 就是 A 的子节点。树的根节点为 1 ，保证树的深度不超过 100 ，且每个节点的子节点不超过 5 个。

水饺很想知道其中最大的果子有多大，于是起床帮忙数遍了 N 个果子，给出了答案。但是水饺不满意果子的大小，它想让果子更大一些，于是使用了一个魔法。这个魔法有两个整数 A,B 构成，可以让节点 A 和 A 的所有子孙节点上果子的大小增加 B ，（节点 A 的子孙节点包括 A 的所有子节点，以及子节点的子节点，以及子节点的子节点的子节点...）。

每次使用完魔法后，水饺都要起床再去找出最大果子的大小，这可把起床累坏了，希望你能写一个程序，帮助起床找到最大值。

input

输入第一行有两个整数， $N,M,(1 \leq N \leq 100000, 1 \leq M \leq 100000)$ ，表示树上节点的个数，以及水饺施放魔法的次数。接下来 N 行,每行两个整数， $f[i],a[i]$ ， $(1 \leq i \leq N, 1 \leq f[i] \leq N, 1 \leq a[i] \leq 100000)$ ，代表第 i 个节点的父节点和一开始第 i 个节点上果子的大小。根节点 1 的父节点为自己。再接下来 M 行，每行两个整数 A,B ， $(1 \leq A \leq N, 0 \leq B \leq 100)$ 表示水饺释放的魔法，含义如题目所说。

output

水饺每次释放魔法后输出一行，包含一个整数，代表当前树上最大果子的大小。

样例输入

[复制](#)

```
5 3
1 1
1 2
1 2
2 4
2 3
3 1
2 2
1 3
```

样例输出

[复制](#)

4

6

9

reversi

一天起床和水饺比赛下黑白棋，水饺持黑棋，起床持白棋，下到一半水饺发现自己处于了劣势，希望起床能让让自己。于是起床和水饺打赌，接下来起床不再下棋，水饺连走五步，如果水饺能够五步内吃掉所有的白棋，就算做水饺赢。黑白棋在8*8的棋盘上进行，输入会给出8*8个字符，表示棋盘，其中'X'表示黑棋，'O'表示白棋，'.'表示空格。落子的主要规则如下：

1. 一步合法的棋步包括：在一个空格新落下一个棋子，并且翻转对手一个或多个棋子。
2. 新落下的棋子与棋盘上已有的同色棋子间，对方被夹住的所有棋子都要翻转过来。可以是横着夹，竖着夹，或是斜着夹。夹住的位置上必须全部是对手的棋子，不能有空格。
3. 一步棋可以在数个方向上翻棋，任何被夹住的棋子都必须被翻转过来，棋手无权选择不翻某个棋子。
4. 除非至少翻转了对手的一个棋子，否则就不能落子。如果一方没有合法棋步，也就是说不管他下到哪里，都不能至少翻转对手的一个棋子，那他这一轮只能弃权。

如果没下过黑白棋对落子规则还不是很明白，可以来此网址试试手，或者百度一下
(http://www.4399.com/flash/66483_3.htm(http://www.4399.com/flash/66483_3.htm))

如果水饺在五步内能够吃掉所有白棋（不一定非要走完五步），就输出"YES"，如果五步内不能吃掉所有白棋，或者过程中无法落子且还有白棋存在，则输出"NO"。

input

输入8行，每行8个字符，其中'X'表示黑棋，'O'表示白棋，'.'表示空格，表示现在棋盘的局势。题目给出的局势一定是一局正常黑白棋游戏能得到的，之后的五步也必须按照正常规则来落子。

output

输出一行，如果水饺在五步内能够吃掉所有白棋（不一定非要走完五步），就输出"YES"，如果五步内不能吃掉所有白棋，或者中间无法落子且还有白棋存在，则输出"NO"。

样例输入1

复制

```
.....
.....
..O.X...
..0000..
.XXXO...
..XX.O..
...X....
.....
```

样例输出1

复制

```
YES
```

样例输入2

复制

```
..X.....
...X.....
..X0XX..
...X0X..
..X0X0X.
.XXX.....
X.X.....
.....
```

样例输出2

复制

```
NO
```