CSGO 模拟赛

题目概况

题目名称	六出祁山	水淹七军	煮酒论英雄	威震逍遥津
程序文件名	climb	graph	cycle	name
输入文件名	climb.in	graph.in	cycle.in	name.in
输出文件名	climb.out	graph.out	cycle.out	name.out
每个测试点时限	1s	1s	2s	1s
运行内存上限	512 MB	512 MB	512 MB	512 MB
结果比较方式	全文比较	spj	全文比较	全文比较
题目类型	传统题	传统	传统	传统

编译选项：

语言	编译命令
C++	-lm -O2 -std=c++14 -Wl,--stack=2147483647

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++ 中函数 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. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
4. 选手应将各题的源程序放在选手文件夹内，不要建立子文件夹 。
5. 评测使用 Windows 系统，系统为 64 位。
6. 评测机配置：Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz，内存 8G

六出祁山（climb）

> 臣亮五出祁山，未得寸土，负罪非轻。今臣复统全师，再出祁山，誓竭力尽心，剿灭汉贼，克复中原，鞠躬尽瘁，死而后已！

题目描述

诸葛丞相在第六次北伐的路上遇到了 n 座大山排成一排，编号从 1 到 n 。每座山有高度 h_i 。

诸葛丞相精通奇门遁甲之术，能用法术移动山石。一次操作中，丞相选定一座山 $2 \leq i \leq n-1$ ，令 $h_i \leftarrow h_i - 1$ 或 $h_i \leftarrow h_i + 1$ 。同时，操作规定不能把山的高度修改为负数。**（特别注意：操作不能改变第一座山和最后一座山的高度！）**

爬山艰辛劳累，为了保存蜀汉军士的体力，以求兴复汉室，还于旧都，诸葛丞相要求操作结束后任意相邻两座山的高度差绝对值不大于 d 。即 $\forall 1 \leq i < n, |h_i - h_{i+1}| \leq d$ 。求达到这一条件的最小操作次数。如果无解，输出 -1 。

输入格式

第一行两个整数 n 和 d ，分别表示山的个数以及合法的最小高度差。

第二行 n 个整数 h_i ，表示每座山的初始高度。

输出格式

输出一个整数，表示最小的操作次数。若无解，输出 -1 。

样例输入 #1

```
4 2
3 0 6 3
```

样例输出 #1

```
4
```

样例输入 #2

```
3 1
6 4 0
```

样例输出 #2

```
-1
```

数据范围与约定

对于 10% 的数据，$2 \leq n \leq 10, 0 \leq d, h_i \leq 10$

对于 30% 的数据，$2 \leq n \leq 300, 0 \leq d, h_i \leq 300$

对于 100% 的数据，$2 \leq n \leq 300, 0 \leq d, h_i \leq 10^9$

水淹七军（graph）

> 夜半征鼙响震天，襄樊平地作深渊。关公神算谁能及，华夏威名万古传。

题目描述

武圣关羽水淹七军，威震华夏，生擒了曹魏大将庞德和于禁。

关公所用的襄樊地区地图可以看作一张 n 个点 m 条边的无向图**（不保证连通）**。现在关公要放水淹掉襄樊，就要对每条边确定一个方向。使其成为有向图。

为了尽快抓住曹魏士兵，关公希望定向后形成的有向图中，最长路径的长度尽可能小。现在关公将这个艰巨的任务交给你，请你输出最长路径长度的最小值，并输出每条边的方向。（图上一条路径的长度该为路径所经过的边的数量）

输入格式

第一行两个正整数 n 和 m ，分别表示点和边的数量。

之后 m 行，每行两个正整数 x_i, y_i 表示第 i 条无向边所连接的两个点。

保证图上没有自环，**但是不保证图上不存在重边，也不保证连通。**

输出格式

第一行输出一个整数，表示最长路径长度的最小值。

然后输出 m 行，每一行两个整数 u_i, v_i 。表示原来的第 i 条边定向为 $u_i \to v_i$ 。

若方案不唯一，输出任意一组方案。

样例输入 #1

```
3 3
1 2
2 3
3 1
```

样例输出 #1

```
2
1 2
2 3
1 3
```

样例输入 #2

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

样例输出 #2

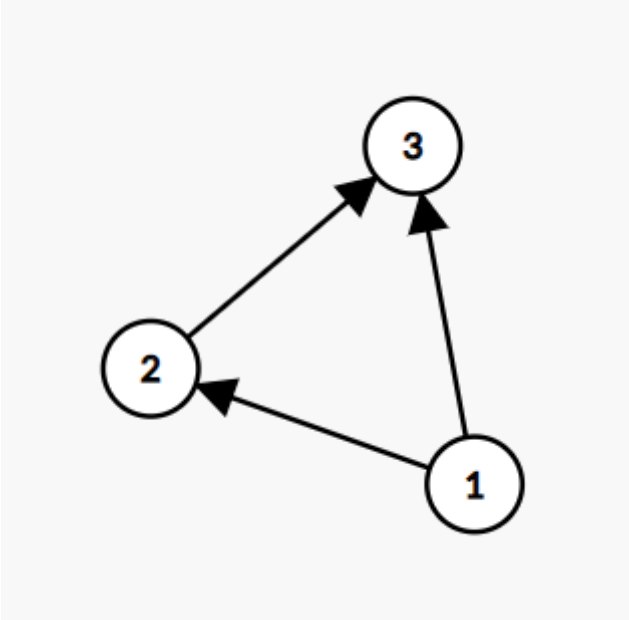
```
3
1 2
2 3
4 3
1 4
1 3
4 2
```

样例解释

样例 #1

最优定向方案如下：

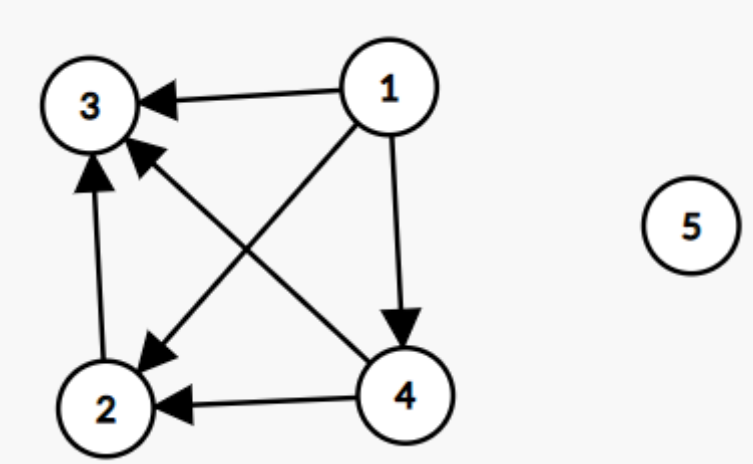
最长路径：$1 \to 2 \to 3$



样例 #2

最优定向方案如下：

最长路径：$1 \to 4 \to 2 \to 3$



数据范围与约定

对于 20% 的数据，$2 \le n \le 10, 0 \le m \le 10$。

对于 100% 的数据，$1 \le n \le 16, 0 \le m \le 1000, 1 \le x_i, y_i \le n$。

煮酒论英雄（cycle）

> “今天下英雄，唯使君与操尔。”

题目描述

曹操与刘备在许都青梅煮酒论英雄，成就千古佳话。有一段野的不能再野的野史记载，曹操为了检验刘备智力是否正常，和刘备玩了这样一个游戏。

曹操的梅子有两种颜色，蓝色的用 B 表示，绿色的用 G 表示。现在曹操取出了若干个梅子**（数量未知，且至少有 2 个）**，在桌上摆成一个环形，这个环上每个梅子都是一个字符 B 或 G 。

阴险狡诈的曹阿瞒把梅子藏起来，然后告诉了刘备 n 条线索。每一条线索是一个字符串 s_i ，由 B 和 G 构成，表示这个环形中，存在连续的一段子串（从一个位置出发，向**顺时针或逆时针方向**走了若干步所经过的所有字符构成的字符串，**可能会绕很多圈**）,这段子串与 s_i 相同。

现在刘备有了这 n 个线索，曹操要求他算出符合这 n 个条件的环形中，最少有几个梅子。刘备身为汉室宗亲，不想给老刘家丢脸，请你帮他写一个程序，输出环形中梅子个数的最小值。

输入格式

第一行包含一个正整数 n ，表示线索个数。

下面 n 行每行一个字符串 s_i ，由 B 和 G 构成。表示第 i 个线索对应的子串。

输出格式

输出一行一个正整数，表示环形中梅子个数的最小值。**（已知梅子个数至少为2！）**

样例输入 #1

```
3
BGGB
BGBGG
GGGBGB
```

样例输出 #1

```
9
```

样例输入 #2

```
2
BGGGBBBGG
GBBBG
```

样例输出 #2

```
6
```

样例解释

样例 #1

最优方案：**GGGBGBGGB** (首尾相连)

从位置 6 开始，向右长度为 4 得到 **BGGB**

从位置 4 开始，向右长度为 5 得到 **BGBGG**

从位置 1 开始，向右长度为 6 得到 **GGBGB**

样例 #2

最优方案：**BGGGBB**（首尾相连）

从位置 1 开始，向右长度为 9 得到 **BGGGBBBGG**

从位置 2 开始，向左长度为 5 得到 **GBBBG**

数据范围与约定

对于 10% 的数据，$n = 1$。

对于 30% 的数据，$1 \le n \le 16, 1 \le |s_i| \le 100$。

对于 100% 的数据，$1 \le n \le 16, 1 \le |s_i| \le 2 \times 10^4$。

威震逍遥津（name）

> 八百铁骑踏江去，十万吴兵丧胆还。虎啸逍遥震千里，江东碧眼犹梦惊！

题目描述

三国时期，孙权多次带兵攻打合肥未果。在逍遥津一役，张辽带领八百名骑兵突破了十万吴兵的防线，杀到孙权帐下。后来江东的小孩子们听见“ 张辽来了”，吓得夜里都不敢啼哭。这就是“张辽止啼”的典故。孙权反思自己亲自领兵送人头的行为之后，决定去安抚东吴的小孩子们。孙权需要给小孩子们起名字。

一个小孩的名字是一个字符串，由 'A'-'Z' 和 'a'-'z' 构成。孙权通过如下规则给小孩取名：

- 初始时，字符串为 "S" 。
- 江东的百姓们制定了 n 个规则，每个规则规定一个大写字母可以转换成一个字符串，例如 'A'->" abCd" 。表示每次操作可以从当前字符串中取出一个大写字母，按照规则将其替换为对应字符串。
- **要求最后得出的字符串不含大写字母。**（大写字母比较大，会让小孩子联想起高大威猛的张辽，所以小孩子们看到大写字母就会被吓掉半条命。）

现在给定 n 个规则和一个整数 l ，求通过任意次变换得到的长度恰好为 l 的字符串中，字典序最小的，并输出该字符串。

上下文无关文法的定义

首先，给出这样一个上下文无关文法的例子：

$$\begin{cases} \text{表示集合：} \{0^n1^n | n \geq 1\} \\ \text{终止集合} = \{0, 1\} \\ \text{Variables} = \{S\} \\ \text{Start symbol} = S \\ \text{Products} = S \to 01, S \to 0S1 \end{cases}$$

语言的文法性描述包括四个重要部分:

1. 一个符号的有穷集合，它构成了被定义语言的串。在上面的示例中该集合为 $\{0, 1\}$,这个字母表称为终结符或终结符号。
2. 一个变元的有穷集合，变元有时也称为非终结符或语法范畴。每个变元代表一个语言，即一个串的集合。在上面的例中只有一个变元 P，它被用来代表以 $\{0, 1\}$ 为字母表的类。
3. 有一个变元称为初始符号，它代表语言开始被定义的地方。在上文中初始符号为 S 。其他变元代表其他辅助的字符串类，这些变元被用来帮助初始符号定义该语言的。
4. 一个产生式（或者规则）的有穷集合，它用来表示语言的递归定义，每个产生式包括:
    1. 一个变元，它被该产生式定义或者部分定义，这个变元通常称为产生式的头。
    2. 一个产生式符号 $\to$ 。
    3. 一个包含零个或多个终结符号或变元的串，它叫作产生式的体，表示一种构成产生式头变元的语言中的串的方法。具体的构造过程是:保持终结符号不动，把任何已知属于该语言的串里出现的产生式的头用产生式的体替换。

上面给出的四个部分构成了一个上下文无关文法，简称文法或者 CFG。其中的 G 可以用组成它的四部分表示，记做 $G = (V, I, P, S)$,其中 V 是变元(Variable)的集合, T 是终结符号(Terminal)的集合, P 是产生式(Production) 的集合，S 代表初始符号(Start symbol)。

在应用一个产生式进行推导时，前后已经推导出的部分结果就是上下文。上下文无关的意思是，只要文法的定义里有某个产生式，不管一个非终结符前后的串是什么，就可以应用相应的产生式进行推导。（从形式上来看，就是产生式的左边都是单独一个非终结符，即形如 $S \to \ldots$，而不是非终结符左右还有别的东西，例如 $aSb \to \ldots$）

上下文无关文法：

产生式：

$$\begin{cases} Sent \to SVO \\ S \to 人|天 \\ V \to 吃|下 \\ O \to 雨|雪|饭|肉 \end{cases}$$

其中英文字母都是非终结符（SVO 分别表示主谓宾），汉字都是终结符。

这个文法可以生成如下句子（共 $2 \times 2 \times 4 = 16$ 种组合）：

$\{$人吃饭，天下雨，人吃肉，天下雪，人下雪，天下饭，天吃肉，$\ldots\ldots\}$

可以看到，其中有一些搭配在语义上是不恰当的，例如 天吃肉。其（最左）推导过程为：

$Sent \to SVO \to 天VO \to 天吃O \to 天吃肉$

但是上下文无关文法里，因为有 $V \to 吃|下$ 这样一条产生式，V 就永远都可以推出 吃 这个词，它并不在乎应用 $V \to 吃|下$ 这个产生式进行推导时 V 所在的上下文（在这个例子里，就是 天VO 中 V 左右两边的字符串 天 和 O）。事实上，在 V 推出 吃 这一步，它的左边是 天 这个词，而 天 和 吃 不搭配，导致最后的句子读起来很奇怪。

上下文有关文法：

$$\begin{cases} Sent \to SVO \\ S \to 人|天 \\ 人V \to 人吃 \\ 天V \to 天下 \\ 下O \to 下雨|下雪 \\ 吃O \to 吃饭|吃肉 \end{cases}$$

可以看到，这里对 V 的推导过程施加了约束：虽然 V 还是能推出 吃 和 下 两个词，但是仅仅当 V 左边是 人 时，才允许它推导出 吃；而当 V 左边是 天 时，允许它推导出 下。这样通过上下文的约束，就保证了主谓搭配的一致性。类似地，包含 O 的产生式也约束了动宾搭配的一致性。

这样一来，这个语言包含的句子就只有 $\{$人吃饭，天下雨，人吃肉，天下雪$\}$ 这四条，都是语义上合理的。以 人吃饭 为例，推导过程为：

$Sent \to SVO \to 人VO \to 人吃O \to 人吃饭$

输入格式

第一行两个整数 n, l，表示规则的个数和目标字符串的长度。

接下来 n 行每行输入一个大写拉丁字母 c_i，后面一个等号 "="，再后面一个字符串 s_i。表示第 i 条规则为 $c_i \to s_i$

所有 s_i 由大写和小写的拉丁字母构成（s_i **可以为空**）。

输出格式

输出一个长度为 l 的字符串，表示能通过任意次操作得到的字典序最小的、长度为 l 的字符串。

如果不存在长度为 l 的字符串，输出 “-” （双引号不输出）。

特别提醒：当答案为空串时，输出一个空行。

样例输入 #1

```
4 3
A=a
A=
S=ASb
S=Ab
```

样例输出 #1

```
abb
```

样例输入 #2

```
4 5
A=aB
A=b
B=SA
S=A
```

样例输出 #2

```
aabbb
```

样例解释

样例 #1

"S" $\rightarrow$ "ASb" $\rightarrow$ "AAbb" $\rightarrow$ "aAbb" $\rightarrow$ "abb"

样例 #2

"S" $\rightarrow$ "A" $\rightarrow$ "aB" $\rightarrow$ "aSA" $\rightarrow$ "aAA" $\rightarrow$ "aaBb" $\rightarrow$ "aaSAb" $\rightarrow$ "aaAAb" $\rightarrow$ "aabbb"

数据范围与约定

对于 10% 的数据，$|s_i| = 1$ 。

对于另外 20% 的数据，$1 \leq n \leq 5$。

对于 100% 的数据，$1 \leq n \leq 50, 0 \leq l, |s_i| \leq 20$。