FJWC 20190212

送分向练习赛 by E.Space

时间：2019 年 2 月 12 日 13:00 ~ 16:00

题目名称	全连	原样输出	不同的缩写
题目类型	传统型	传统型	传统型
目录	`fc`	`copy`	`diff`
可执行文件名	`fc`	`copy`	`diff`
输入文件名	`fc.in`	`copy.in`	`diff.in`
输出文件名	`fc.out`	`copy.out`	`diff.out`
每个测试点时限	1.0 秒	3.0 秒	1.0 秒
内存限制	256 MB	256 MB	256 MB
测试点/包数目	20	10	3
测试点是否等分	是	是	否

提交源程序文件名

对于 C++ 语言	`fc.cpp`	`copy.cpp`	`diff.cpp`
对于 C 语言	`fc.c`	`copy.c`	`diff.c`
对于 Pascal 语言	`fc.pas`	`copy.pas`	`diff.pas`

编译选项

对于 C++ 语言	`-m32`
对于 C 语言	`-lm -m32`
对于 Pascal 语言	

全连（fc）

【题目背景】

还记得若干年前那段互相比较《克罗地亚狂想曲》的分数的日子吗？

【题目描述】

E.Space 喜欢打音游。

但是他技术不好，总是拿不到全连（Full Combo）。

现在他面前有一份乐谱，乐谱的其中一段有 n 个连续的单键音符。

相邻两个音符的到来时间均相等，我们可以认为第 i 个音符会在第 i 个时刻到来。

点击一个音符，E.Space 需要一段准备时间来进行移动手指之类的操作。由于音符的位置和周围情况不同，点击每个音符的准备时间也不同。

在一个音符的准备时间内，E.Space 没法做到去点击其它音符，但是不同音符的准备时间范围可以互相重叠。形式化地，令第 i 个音符的准备时间为 t_i 个单位时间，那么如果 E.Space 选择去点击第 i 个音符，那么他就没法点击所有到来时刻在 $(i-t_i, i+t_i)$ 中的音符。

为了获得更高的分数，E.Space 还计算了每个音符的性价比。一个音符的性价比等于点击这个音符得到的分数除以 E.Space 点击它所需要的准备时间。

E.Space 就不指望全连了，他只是想让你帮他计算一下他最多可以得到多少分数。

【输入格式】

从文件 ***fc.in*** 中读入数据。

第一行一个正整数 n 。

第二行 n 个正整数，第 i 个正整数表示 t_i 。

第三行 n 个正整数，第 i 个正整数表示第 i 个音符的性价比 a_i。

【输出格式】

输出到文件 ***fc.out*** 中。

一行一个正整数，表示 E.Space 可能达到的最高分数。

【样例 1 输入】

```
5
2 3 2 1 2
3 1 2 9 4
```

【样例 1 输出】

```
18
```

【样例 1 解释】

E.Space 可以选择点击第 $1,3,5$ 个音符，分数为 $2\times3+2\times2+2\times4=18$ 。

【子任务】

保证 $t_i \le n$ ， $a_i \le 10^9$

测试点编号	$n \le$
1	5
2	10
3	15
4	20
5	1000
6	2000
7	5000
8	10000
9	30000
10	50000
11	100000
12	200000
13	500000
14	800000
15	1000000
16	1000000
17	100000
18	100000
19	1000000
20	1000000

对于最后 4 个测试点，保证对于任意的 i,j 有 $t_i=t_j$ 。

原样输出（copy）

【题目描述】

nealchen 是一只 copycat。

它会把输入按行读入，原封不动地复制到输出中去。

但是在一次更新以后，它的程序出了一些问题。

它没法输出换行符了。

并且，读入的时候，总会莫名其妙地随机漏掉开头和结尾的若干个字符，甚至整行都会漏掉。

比如 orznight 可能会变成 rzni，orz，h 或者空串。

现在你找到一份输入文件丢给 nealchen，你想知道它的输出可能有多少种情况，以及每种情况分别是什么。

由于你找到的输入文件全部来自之前的福建省选，所以所有的输入文件每行只可能包含 A，C，G，T 四种字符。

【输入格式】

从文件 ***copy.in*** 中读入数据。

第一行一个正整数 n，表示（题面中）输入文件的行数。

接下来 n 行，表示输入文件的内容。保证这 n 行中每行的每个字符是 A，C，G，T 四种字符中的一种。

接下来一个整数 $k(0 \le k \le 1)$，具体含义详见输出格式。

【输出格式】

输出到文件 ***copy.out*** 中。

若 $k = 0$，你需要输出一行，表示输出的可能情况个数模 $10^9 + 7$ 的结果。

若 $k = 1$，你需要按照字典序从小到大输出所有可能的输出情况，一行一个字符串，最后一行输出输出的可能情况个数模 $10^9 + 7$ 的结果。

【样例 1 输入】

```
3
AC
CC
AA
0
```

【样例 1 输出】

```
22
```

【样例 2 输入】

```
3
AC
CC
AA
1
```

【样例 2 输出】

```
A
AA
AAA
AC
ACA
ACAA
ACC
ACCA
ACCAA
ACCC
ACCCA
ACCCAA
C
CA
CAA
CC
CCA
CCAA
CCC
CCCA
CCCAA
22
```

【样例 2 解释】

注意输出的第一行是一个空行。

【子任务】

对于 40% 的数据，$n = 1$

对于 60% 的数据，$n < 3$

对于 100% 的数据，保证输入文件大小不超过 1MB ，保证输出文件大小不超过 200MB 。

不同的缩写（diff）

【题目描述】

你在写一款 Galgame 的剧情（的代码）。

在这个游戏中一共有 n 个角色。你需要编写一些关于这些角色的对话内容。然而，在写这些对话内容之前，都要写一段关于角色信息的代码，就像这样：

```
Character("Alex", color = "#FFFC3A")
```

你觉得这样好麻烦。你决定把它简化一下。你打算用角色名字的一个子序列（可以不连续）来作为它的简称。

当然，不同的角色要用不同的字符串作为简称，否则你就变量重名了。

你想确定一个简称的分配方案使得所有角色中最长的简称尽量短，这样你打起代码就会方便一些。

【输入格式】

从文件 ***diff.in*** 中读入数据。

第一行一个正整数 n 。

接下来 n 行，每行一个由小写字母组成的字符串，代表一个角色的名字。

不同的角色可能会有相同的名字。

【输出格式】

输出到文件 ***diff.out*** 中。

如果不存在一种分配简称的方案满足条件，输出 -1 。

否则第一行输出一个正整数，表示最长的简称的最小长度。

接下来 n 行每行一个字符串，按顺序表示每个角色的简称。

若有多种方案满足条件，那么你可以输出任意一种。

【样例输入】

```
11
night
nealchen
beimingyouyu
echo
rankinf
dntcrybecthlev
lagoon
```

```
cyc
alphagem
leehwincing
clin
```

【样例输出】

```
1
g
a
m
e
r
b
o
y
l
w
c
```

【子任务】

保证 $n \le 300$ ，每个名字的长度不超过 300 。

【子任务 1（30 pts）】

$n \le 4$

【子任务 2（30 pts）】

$n \ge 100$
串长和串的内容在题目范围内均匀随机。
即串长在 $[1,300]$ 内随机，串的每一位在 a 到 z 之间随机。

【子任务 3（40 pts）】

无特殊限制