A. Hot Bath

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bob is about to take a hot bath.

There are two taps to fill the bath: a hot water tap and a cold water tap. The cold water's temperature is t_1, and the hot water's temperature is t_2. The cold water tap can transmit any integer number of water units per second from 0 to x_1, inclusive. Similarly, the hot water tap can transmit from 0 to x_2 water units per second.

If y_1 water units per second flow through the first tap and y_2 water units per second flow through the second tap, then the resulting bath water temperature will be:

$$t = \frac{t_1 y_1 + t_2 y_2}{y_1 + y_2}$$

Bob wants to open both taps so that the bath water temperature was not less than t_0. However, the temperature should be as close as possible to this value. If there are several optimal variants, Bob chooses the one that lets fill the bath in the quickest way possible.

Determine how much each tap should be opened so that Bob was pleased with the result in the end.

Input

You are given five integers t_1, t_2, x_1, x_2 and t_0 ($1 \le t_1 \le t_0 \le t_2 \le 10^6$, $1 \le x_1, x_2 \le 10^6$).

Output

Print two space-separated integers y_1 and y_2 ($0 \le y_1 \le x_1$, $0 \le y_2 \le x_2$).

Sample test(s)

input
```
10 70 100 100 25
```
output
```
99 33
```

input
```
300 500 1000 1000 300
```
output
```
1000 0
```

input
```
143 456 110 117 273
```
output
```
76 54
```

Note

In the second sample the hot water tap shouldn't be opened, but the cold water tap should be opened at full capacity in order to fill the bath in the quickest way possible.

B. Password

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Asterix, Obelix and their temporary buddies Suffix and Prefix has finally found the Harmony temple. However, its doors were firmly locked and even Obelix had no luck opening them.

A little later they found a string s, carved on a rock below the temple's gates. Asterix supposed that that's the password that opens the temple and read the string aloud. However, nothing happened. Then Asterix supposed that a password is some substring t of the string s.

Prefix supposed that the substring t is the beginning of the string s; Suffix supposed that the substring t should be the end of the string s; and Obelix supposed that t should be located somewhere inside the string s, that is, t is neither its beginning, nor its end.

Asterix chose the substring t so as to please all his companions. Besides, from all acceptable variants Asterix chose the longest one (as Asterix loves long strings). When Asterix read the substring t aloud, the temple doors opened.

You know the string s. Find the substring t or determine that such substring does not exist and all that's been written above is just a nice legend.

Input

You are given the string s whose length can vary from 1 to 10^6 (inclusive), consisting of small Latin letters.

Output

Print the string t. If a suitable t string does not exist, then print "`Just a legend`" without the quotes.

Sample test(s)

input
```
fixprefixsuffix
```
output
```
fix
```

input
```
abcdabc
```
output
```
Just a legend
```

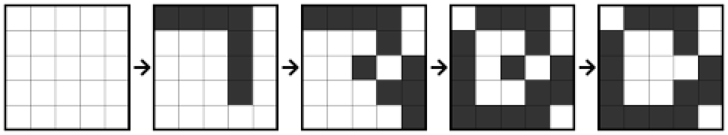
C. E-reader Display

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After years of hard work scientists invented an absolutely new e-reader display. The new display has a larger resolution, consumes less energy and its production is cheaper. And besides, one can bend it. The only inconvenience is highly unusual management. For that very reason the developers decided to leave the e-readers' software to programmers.

The display is represented by $n \times n$ square of pixels, each of which can be either black or white. The display rows are numbered with integers from 1 to n upside down, the columns are numbered with integers from 1 to n from the left to the right. The display can perform commands like "x, y". When a traditional display fulfills such command, it simply inverts a color of (x, y), where x is the row number and y is the column number. But in our new display every pixel that belongs to at least one of the segments (x, x) - (x, y) and (y, y) - (x, y) (both ends of both segments are included) inverts a color.

For example, if initially a display 5×5 in size is absolutely white, then the sequence of commands $(1, 4)$, $(3, 5)$, $(5, 1)$, $(3, 3)$ leads to the following changes:



You are an e-reader software programmer and you should calculate minimal number of commands needed to display the picture. You can regard all display pixels as initially white.

Input

The first line contains number n $(1 \le n \le 2000)$.

Next n lines contain n characters each: the description of the picture that needs to be shown. "0" represents the white color and "1" represents the black color.

Output

Print one integer z — the least number of commands needed to display the picture.

Sample test(s)

input

```
5
01110
10010
10001
10011
11110
```

output

```
4
```

D. Fibonacci Sums

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fibonacci numbers have the following form:

$$F_1 = 1,$$
$$F_2 = 2,$$
$$F_i = F_{i-1} + F_{i-2}, i > 2.$$

Let's consider some non-empty set $S = \{s_1, s_2, ..., s_k\}$, consisting of **different** Fibonacci numbers. Let's find the sum of values of this set's elements:

$$\sum_{i=1}^{k} s_i = n$$

Let's call the set S a number n's *decomposition into Fibonacci sum*.

It's easy to see that several numbers have several decompositions into Fibonacci sum. For example, for 13 we have $13, 5 + 8, 2 + 3 + 8$ — three decompositions, and for 16: $3 + 13, 1 + 2 + 13, 3 + 5 + 8, 1 + 2 + 5 + 8$ — four decompositions.

By the given number n determine the number of its possible different decompositions into Fibonacci sum.

Input

The first line contains an integer t — the number of tests $(1 \le t \le 10^5)$. Each of the following t lines contains one test.

Each test is an integer n $(1 \le n \le 10^{18})$.

Please do not use the `%lld` specificator to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specificator.

Output

For each input data test print a single number on a single line — the answer to the problem.

Sample test(s)

input

```
2
13
16
```

output

```
3
4
```

Note

Two decompositions are different if there exists a number that is contained in the first decomposition, but is not contained in the second one. Decompositions that differ only in the order of summands are considered equal.

E. Pills

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Doctor prescribed medicine to his patient. The medicine is represented by pills. Each pill consists of a shell and healing powder. The shell consists of two halves; each half has one of four colors — blue, red, white or yellow.

The doctor wants to put 28 pills in a rectangular box 7×8 in size. Besides, each pill occupies exactly two neighboring cells and any cell contains exactly one half of a pill. Thus, the result is a four colored picture 7×8 in size.

The doctor thinks that a patient will recover sooner if the picture made by the pills will be special. Unfortunately, putting the pills in the box so as to get the required picture is not a very easy task. That's why doctor asks you to help.

Doctor has some amount of pills of each of 10 painting types. They all contain the same medicine, that's why it doesn't matter which 28 of them will be stored inside the box.

Place the pills in the box so that the required picture was formed. If it is impossible to place the pills in the required manner, then place them so that the number of matching colors in all 56 cells in the final arrangement and the doctor's picture were maximum.

Input

First 7 lines contain the doctor's picture. Each line contains 8 characters, each character can be "B", "R", "W" and "Y" that stands for blue, red, white and yellow colors correspondingly.

Next four lines contain 10 numbers that stand for, correspondingly, the number of pills painted:

"BY" "BW" "BR" "BB"

"RY" "RW" "RR"

"WY" "WW"

"YY"

Those numbers lie within range from 0 to 28 inclusively. It is guaranteed that the total number of pills in no less than 28.

Output

Print on the first line the maximal number cells for which the colors match.

Then print 13 lines each containing 15 characters — the pills' position in the optimal arrangement. The intersections of odd lines and odd columns should contain characters "B", "R", "W" and "Y". All other positions should contain characters ".", "-" and "|". Use "-" and "|" to show which halves belong to one pill. See the samples for more clarification.

If there are several possible solutions, print any of them.

Sample test(s)

input

```
WWWBBWWW
WWWBBWWW
YYWBBWWW
YYWBBWRR
YYWBBWRR
YYWBBWRR
YYWBBWRR
0 0 0 8
0 1 5
1 10
5
```

output

```
53
W.W.W.B.B.W.W.W
|.|.|.|.|.|.|.|
W.W.W.B.B.W.W.W
...............
Y.Y.W.B.B.W.W-W
|.|.|.|.|.|....
Y.Y.W.B.B.W.R.R
............|.|
Y.Y.W.B.B.R.R.R
|.|.|.|.|.|....
Y.Y.W.B.B.W.R.R
............|.|
Y-Y.B-B.B-B.R.R
```

input

```
WWWWWWWW
WBYWRBBY
```

```
BRYRBWYY
WWBRYWBB
BWWRWBYW
RBWRBWYY
WWWWWWWW
0 0 0 1
0 0 1
0 1
25
```

output

```
15
W.Y.Y-Y.Y-Y.Y-Y
|.|............
W.Y.Y.Y.Y.B-B.Y
....|.|.|.....|
Y-Y.Y.Y.Y.Y-Y.Y
...............
Y.Y.Y.R.Y.Y.Y-Y
|.|.|.|.|.|....
Y.Y.Y.R.Y.Y.Y.Y
............|.|
Y-Y.Y.Y-Y.Y.Y.Y
....|.....|....
Y-Y.Y.Y-Y.Y.Y-Y
```