

Problems

Problem	ICONS	DOMAINS	CLOCK	EXCHANGE
Executable file	ICONS.EXE	DOMAINS.EXE	CLOCK.EXE	EXCHANGE.EXE
Source file	ICONS.BAS	DOMAINS.BAS	CLOCK.BAS	EXCHANGE.BAS
	ICONS.PAS	DOMAINS.PAS	CLOCK.PAS	EXCHANGE.PAS
	ICONS.C	DOMAINS.C	CLOCK.C	EXCHANGE.C
	ICONS.CPP	DOMAINS.CPP	CLOCK.CPP	EXCHANGE.CPP
Input file	ICONS.IN	DOMAINS.IN	CLOCK.IN	EXCHANGE.IN
Output file	ICONS.OUT	DOMAINS.OUT	CLOCK.OUT	EXCHANGE.OUT
Time limit per test	10 seconds	10 seconds	10 seconds	10 seconds
Number of tests	10	10	10	10
Points per test	3	4	6	7
Total points	30	40	60	70
	200			

ICONS

Dave has a collection of N interesting pebbles. He wishes to arrange them in rows and columns in such a way that the sum of number of rows and number of columns needed is minimal possible. Write a program that will help Dave to find such numbers.

Input data

The first and only line of input file contains a natural number N ($1 \leq N \leq 100$), the number of pebbles to be arranged. Arrangement needs not to be regular in any sense – some places in a row may be empty.

Output data

The first and only line of output file must contain number of rows and number of columns separated by one space.

Note: A solution needs not to be unique.

Examples

ICONS.IN	ICONS.IN	ICONS.IN
2	5	14
ICONS.OUT	ICONS.OUT	ICONS.OUT
1 2	3 2	4 4

DOMAINS

Dave likes to surf on Internet and visit many interesting web sites. He stores addresses of web pages he visited in a file.

Make a program that will help Dave to determine the most often visited domains. Also, the program should find how often Dave visited them.

Input data

The first line of input file contains a natural number N , $1 \leq N \leq 100$, the number of addresses.

Each of the following N lines contains one simplified address. A simplified address **may** begin with a prefix '**http://**', after which comes one or more words separated with dot ('.'). That part of an address represents a name of a computer. After that **may** come a path consisting of character '/' followed by one or more words also separated with character '/'. Each word is a sequence of one or more small letters of English alphabet (a–z). Domain name is the last word in a name of a computer and consists of three characters at most.

Every address will consist of 50 characters at most.

Output data

The first line of output file should contain the number of visits to the most often visited domain(s).

The second line should contain all the most often visited domains in any order. Every two domains should be separated by one space character.

Examples

DOMAINS.IN

```
3
www.hr/www
com
http://abc.def.net
```

DOMAINS.OUT

```
1
com hr net
```

DOMAINS.IN

```
4
http://abc.def
http://abc.def.ghi
def.ghi/abc
http://bea.vis.but.thead.xyz
```

DOMAINS.OUT

```
2
ghi
```

DOMAINS.IN

```
5
com/put1/put2
http://def.exe
my.dad.exe/put
nesto.bat
http://my.mom.com
```

DOMAINS.OUT

```
2
exe com
```

CLOCK

One day, Dave made a digital clock. Each of four numbers needed to display a time (in hhmm format) on it is represented with 15 small red LED diodes arranged in a 5x3 matrix as shown below ('#' represents lit diode, and '.' unlit diode).

```
###   . .#   ###   ###   #.#   ###   ###   ###   ###   ###  
#.#   . .#   . .#   . .#   #.#   #..   #..   . .#   #.#   #.#  
#.#   . .#   ###   ###   ###   ###   ###   . .#   ###   ###  
#.#   . .#   #..   . .#   . .#   . .#   #.#   . .#   #.#   . .#  
###   . .#   ###   ###   . .#   ###   ###   . .#   ###   ###
```

Dave set the correct time on his new clock and, being exhausted with effort needed to make the clock went to sleep.

When he woke up next morning, he looked at his new clock and found out that some of diodes went off.

Write a program that will determine the earliest possible time that corresponds to Dave's faulty clock display.

Input data

Input file contains four 5x3 groups of characters '#' and '.' separated by one column of spaces. Those four groups represent arrangement of lit and unlit diodes of a faulty display.

Output data

The first and only line of output file should contain desired time.

Examples

CLOCK.IN

```
#.#   . . .   . . .   #..  
#.#   . . .   . . .   #..  
#.#   ###   ###   ###  
#.#   #..   . .#   . .#  
###   ###   ###   . .#
```

CLOCK.OUT

02:34

CLOCK.IN

```
###   ###   . .#   . .  
. . .   #.#   #.#   #.#  
###   #.#   .#   .#  
#..   #.#   . .#   . .#  
###   #.#   . .#   ###
```

CLOCK.OUT

20:48

CLOCK.IN

```
#..   #.#   #.#   .#  
. .#   #..   . . .   #..  
#.#   .##   ###   .##  
. .#   #.#   . . .   . .  
#.#   ###   #.#   ###
```

CLOCK.OUT

06:25

EXCHANGE

Dave somehow acquired exchange rates of US dollar to German marks for several days in the future. Write a program that will suggest Dave when to buy or sell marks or dollars so that, starting with 100 marks he ends up with the highest possible amount of marks at the end of the last day.

Input data

The first line of input file contains a natural number N , $1 \leq N \leq 100$, the number of future days for which Dave knows exchange rates.

Each of N following lines contains two natural numbers B and S separated with a space, $100 \leq B \leq S \leq 1000$. The $(i+1)^{\text{th}}$ line describes the exchange rate of i^{th} day in advance.

Those two numbers determine the exchange rate in the following way: B dollars can be bought for 100 marks, and 100 marks can be bought for S dollars.

Output data

The first and only line of output file should contain a desired amount written to two decimal places.

Remark: Due to real arithmetic round off errors, results within error of 0.05 marks of correct value will be regarded as correct.

Examples

EXCHANGE.IN

3
393 398
394 401
386 386

EXCHANGE.OUT

102.07

EXCHANGE.IN

5
300 300
310 320
320 330
330 330
300 320

EXCHANGE.OUT

103.12

EXCHANGE.IN

8
218 219
228 231
227 235
205 213
230 232
239 239
251 258
205 213

EXCHANGE.OUT

126.14

Explanation (needs not to be printed)

Example #3

Day 2 ... changing 100.0000 DEM to 228.0000 USD
Day 4 ... changing 228.0000 USD to 107.0422 DEM
Day 7 ... changing 107.0422 DEM to 268.6760 USD
Day 8 ... changing 268.6760 USD to 126.1389 DEM