Caesar Cypher (caesar)

Julius Caesar has just written the new war declaration against Sparta. To prevent the enemies from reading this precious document the Emperor has ordered to encrypt it using his cypher.

> ☞ The Caesar Cypher consists of changing each letter of the message with next k-th letter of the alphabet (wrapping to a after z). k is called the *key* of the cypher and it can be used to decrypt the message simply by re-encrypting it with $-k$ as the key.
>
> For example if the message was `pizza` and the key $k = 3$ the encrypted message would be `slccd`.

Unfortunately the official who was supposed to encrypt the messages messed it up. He put the war declaration messages with other unrelated messages. Luckily for him he noticed that if he splits the messages into piles all with the same message (maybe encrypted with different keys), the taller is the one!

Help him to find the size of the largest pile all with the same message.

> ☞ Among the attachments of this task you may find a template file `caesar.*` with a sample incomplete implementation.

Input

The first line contains two integers N and D, where N is the number of messages, D is the length of each message. The next N lines contain a single word each.

Output

You need to write a single line with an integer: the size of the largest pile.

Constraints

- $1 \leq N \leq 10\,000$.
- $1 \leq D \leq 1000$.
- Each message is a single word with only lower case ASCII letters.

Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases. These test cases are such that:

- in 50% of them, $N \leq 1000$,

Examples

stdin/input.txt	stdout/output.txt
5 4 aaaa bbbb abab fgfg hhhh	3
5 5 pizza mafia cacca puzza pizzo	1

Explanation

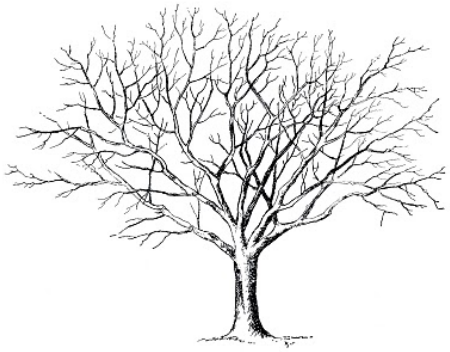
In the **first example** there are 2 piles:

- `aaaa`, `bbbb` and `hhhh`
- `abab`, `fgfg`

The largest is the first so the answer is 3. Note that, assuming `aaaa` is the unencrypted message, `bbbb` can be decrypted with $k = 1$ and `hhhh` with $k = 7$.

In the **second example** each message is unique and cannot be transformed into any others. So there are 5 piles each with one message.

Wonderful Grove (copac)



Silviu the lumberjack, who lives in "The Wonderful Grove", is well known for his ability and is ready to tackle another challenging task. Winter is coming and a big tree needs to be cut.

Silviu is carefully studying the structure of the tree, taking precise notes about each branch and its connections. He has labelled all the points of connection between two or more branches with numbers from 1 to N.

He has opted for making a single cut (after all, every cut requires physical effort...), getting rid of the branches conjuncting there and transporting the remaining smaller pieces. The company selected for transport is going to accept a part of the tree only if it is smaller (or equal) than half of the size of the original tree. The size of a (part of a) tree is merely the number of the conjunction points in it.

At which points can Silviu make his cut to satisfy the requirement from the transport company?

> ☞ Among the attachments of this task you may find a template file `copac.*` with a sample incomplete implementation.

Input

The first line contains the only integer N. The following $N-1$ lines contain two integers X_i and Y_i each, meaning that there is a branch between those two points.

Output

You need to write a line for every conjuction point that can be chosen for the cut. If multiple points respect the constraint, they must be sorted in ascending order.

Constraints

- $1 \leq N \leq 100\,000$.
- $1 \leq X_i, Y_i \leq N$ for each $i = 0 \ldots N-2$.
- It is guaranteed that at least one answer exists.

Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases.

Examples

stdin/input.txt	stdout/output.txt
7 1 2 1 3 2 4 3 5 3 6 3 7	3

Explanation

In the **sample case** the only feasible solution is to make a cut at point 3. In that way the original tree is divided in four pieces (one of size 3 and three of size 1).

Decimal Fractions (`fraction`)

Archimedes has just discovered a new concept: the *fraction*, where an integer number (numerator) is divided by another integer number (denominator). In particular, he is now focusing on the simplest kind of fractions, called *unit fraction*, where the numerator is exactly 1.

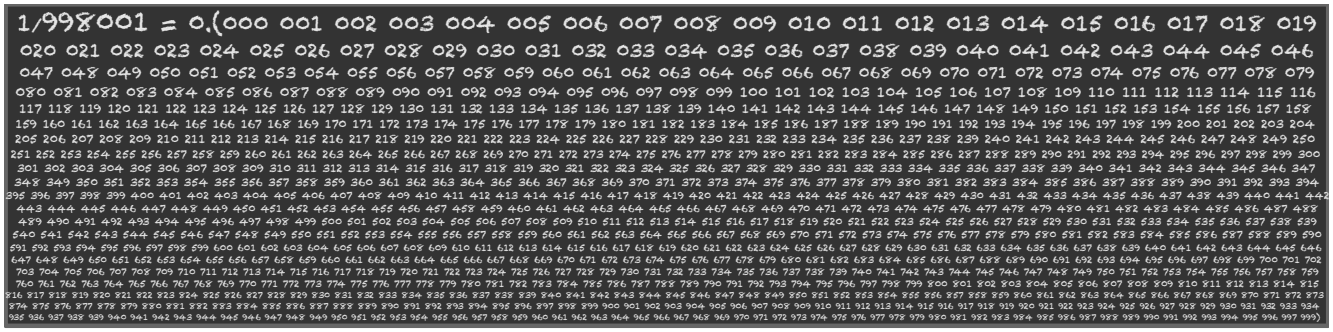


Figure 1: A remarkable unit fraction: every three-digit number is in the cycle... except for 998, of course.

Until now, Archimedes managed to compute the decimal representation of all the unit fractions with denominator up to ten, and noticed that some of them contain a *recurring cycle* of digits.

```
1/2 = 0.5
1/3 = 0.(3)
1/4 = 0.25
1/5 = 0.2
1/6 = 0.1(6)
1/7 = 0.(142857)
1/8 = 0.125
1/9 = 0.(1)
1/10 = 0.1
```

Where `0.1(6)` means `0.166666...`, and has a 1-digit recurring cycle. Archimedes is particularly excited by long recurring cycles, like the one of $1/7$ which is 6 digits long, and would like to discover an even longer one with denominator up to N; but is getting tired of computing all those decimals. Help Archimedes in his quest, by finding the largest number $d < N$ such that $1/d$ contains the longest possible recurring cycle in its decimal fraction part!

> ☞ Among the attachments of this task you may find a template file `fraction.*` with a sample incomplete implementation.

Input

The first and only line contains a single integer N.

Output

You need to write a single line with an integer: the largest $d < N$ with a longest recurring cycle.

Constraints

- $2 \leq N \leq 1\,000\,000$.

Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases. These test cases are such that:

- in 50% of them, $N \leq 1000$,
- in 80% of them, $N \leq 30\,000$.

Examples

stdin/input.txt	stdout/output.txt
10	7
20	19

Word Game (jumps)

James Joyce is composing a new mind-blowing masterpiece. In his next work, two words w_1 and w_2 will be allowed to be adjacent only if they are *close enough*. We define w_2 to be close enough to w_1 (or that w_1 *jumps* to w_2) if one of the following conditions is verified:

- w_2 is obtained from w_1 by deleting one letter.
- w_2 is obtained from w_1 by replacing one of the letters in w_1 by some letter that appears to its right in w_1 and which is also to its right in alphabetical order.

For example, we can jump from `abbca` to `abca` by deleting the second (or the third) letter. We can jump from `aabca` to `abbca` by replacing the `a` in the second position by the letter `b` that appears to the right of the `a` in `aabca` and which is also to its right in alphabetical order. On the other hand, we cannot jump from `abbca` to `aabca` since we would need to replace the `b` in the second position by `a`, but `a` is to the left of `b` in alphabetical order.

You will be given a collection of words W, which James want to put in his next masterpiece. Your task is to find the length of the longest sequence $w_1, w_2, \ldots$ of distinct words from W such that we may jump from w_1 to w_2, w_2 to w_3 and so on. We call this the *jumping number* for this set.

For example, if

$$W = \{abacd, bcdada, dd, abcd, bcdd, adcd, addd, aa, ccd, add, ad\}$$

then, the jumping number is 7 corresponding to the sequence

$$abacd, abcd, adcd, addd, add, ad, dd.$$

☞ Among the attachments of this task you may find a template file `jumps.*` with a sample incomplete implementation.

Input

The first line contains the only integer N indicating the number of words in the input. The following N lines contain one word W_i.

Output

The output should be a single integer, indicating the *jumping number* of the given set of words.

Constraints

- $1 \leq N \leq 100$.
- The length of each word is at most 10.
- Each word is formed by lowercase alphabetic letters (`a-z`).

Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases.

Examples

stdin/input.txt	stdout/output.txt
11 abacd bcdada dd abcd bcdd adcd addd aa ccd add ad	7

Fractal Painting (painting)

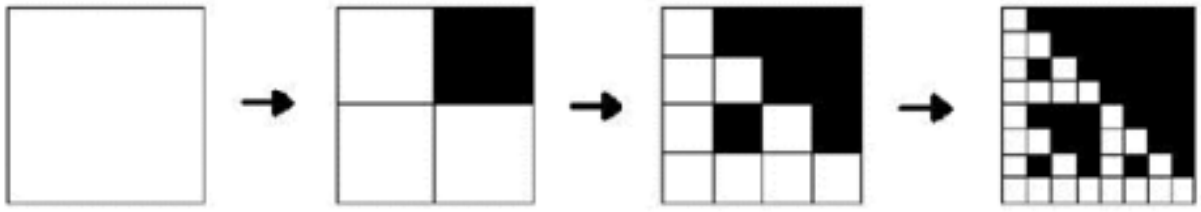
Ever since Marcus (a famous Dutch abstractionist) heard of fractals, he made them the main topic of his canvases. Every morning, the artist takes a piece of graph paper and starts making a model of his future canvas, by selecting a rectangular area of $N \times N$ squares and painting some of these squares black. Then, he takes a clean square piece of paper and paints the fractal using the following algorithm:

1. The paper is divided into N^2 identical squares and some of them are painted black according to the model.
2. Every square that remains white is divided into N^2 smaller squares and some of them are painted black according to the model.

⋮

K. Repeat step 2.

At the end of the process, a canvas consisting of $N^K \times N^K$ black and white squares is produced.



Unfortunately, this tiresome work demands too much time from his painting genius. Marcus has been dreaming of making the process automatic to move to making 3D or even 4D fractals: help him!

☞ Among the attachments of this task you may find a template file `painting.*` with a sample incomplete implementation.

Input

The first line contains integers N, K. Each of the following N lines contains N symbols that determine the model, where:

- '`.`' represents a white square;
- '`*`' represents a black square.

Output

You need to produce an $N^K \times N^K$ matrix representing what a picture should look like after K steps of the algorithm.

Constraints

- $2 \leq N \leq 10$.
- $N^K \leq 17\,000$

Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases. These test cases are such that:

- in 40% of them, $N = 2$,
- in 70% of them, $N \leq 4$.

Examples

stdin/input.txt	stdout/output.txt
2 3 .* ..	.******* ..****** .*.***** ....**** .***.*** ..**..** .*.*.*.* ....
3 2 .*. *** .*.	.*.***.*. ********* .*.***.*. ********* ********* ********* .*.***.*. ********* .*.***.*.

Number Game (subset)

Rich Uncle Pennybags is getting tired of playing Monopoly, hence developed a new exciting *number game*. In this game, V tokens are randomly extracted, each depicting a distinct positive integer. Each player then has to select a subset S such that every pair (S_i, S_j) of elements in this subset satisfies: $S_i \% S_j = 0$ or $S_j \% S_i = 0$. The player who picks up the bigger subset wins!

Rich Uncle Pennybags is not used to lose a game: help him find the largest allowed subset!

> ☞ Among the attachments of this task you may find a template file `subset.*` with a sample incomplete implementation.

Input

The first line contains the only integer N indicating the cardinality (number of elements) of the set V. The following line contains N integers, the elements V_i of the set.

Output

The output should be a single integer, indicating the *largest cardinality* of a subset S of V such that the condition is satisfied. If there are multiple solution, output any of them.

Constraints

- $1 \leq N \leq 1000$.
- $V_i < 2^{32}$ for any i.

Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases.

Examples

stdin/input.txt	stdout/output.txt
5 1 2 4 8 5	4

UEFA Coefficient (`uefa`)

Everyone in the world knows Andrew Wiles' addiction to football. Recently, a strange number has been calculated for each country: the "UEFA coeffecient". This coefficient is so important because it determins how many teams for each country are allowed to compete in the world's most prestigious tournament, the UEFA Champions League.

Andrew loves playing with numbers and computers, but he's a bit skeptical about *how* that coefficient is computed by UEFA and would prefer to calculate it himself. He gathered a list of N teams, each with its country, and a list of M results of played matches.

The rules to calculate a team's coefficient are quite straightforward:

- Every team gets *one* point for every match it has played;
- Additional *three* points are awarded to the winner (i.e., the team with the highest number of goals scored). If the match ends with a draw, *one* point is awarded to both teams.

The coefficient associated with a country is the sum of all coefficients of teams belonging to it. Help Andrew writing a program that computes the final ranking, ordered by decreasing coefficient, with countries and their global coefficients!

☞ Among the attachments of this task you may find a template file `uefa.*` with a sample incomplete implementation.

Input

The first line contains integers N and M. The following N lines contain two strings each: T_i and C_i, respectively the name of the i-th team and its country. The subsequent M lines contain two strings (T_{home} and T_{away}) and two integers representing the score of that match (i.e. the number of goals, respectively G_{home} and G_{away}).

Output

You need to write K lines, one for every country. Each of them should contain a string and an integer: respectively, the name of the country and its UEFA coefficient.

Constraints

- $2 \leq N \leq 1\,000$.
- $1 \leq M \leq 10\,000$.
- Each team name T_i and each country C_i is a string of at most 10 lowercase letters of the English alphabet.
- A team never plays a match against itself.
- Integers G_{home} and G_{away}, the number of goals, are both non negative.
- Whenever two countries share exactly the same coefficient, they should be ordered lexicograpically.

Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases. These test cases are such that:

- in 50% of them, there is only one country (i.e., $K = 1$).

Examples

stdin/input.txt	stdout/output.txt
4 4 inter italy milan italy bayern germany psg france inter milan 1 0 bayern inter 4 1 milan inter 2 2 milan psg 1 0	italy 14 germany 4 france 1
3 3 inter italy milan italy juventus italy inter milan 3 3 juventus inter 0 1 inter juventus 0 3	italy 14

Explanation

In the **first sample case** the leading country is Italy with 14 as coefficient, given from 6 points for playing matches, 3 points for Inter winning the first match, 2 points for the draw in the third one, 3 points for Milan winning the fourth match.

In the **second sample case** every team is Italian, thus Italy is the only country appearing in the ranking.