

ACM ICPC 2005–2006 NEERC Moscow Subregional Contest

Moscow, October 23, 2005

List of Problems

Problem	Page	Time limit	Memory limit	Name
A	2	1 second	64 MB	Anti-factorial
B	3	1 second	64 MB	Battery Upgrade
C	4	1 second	64 MB	Control Tower
D	5	1 second	64 MB	Dinner Time
E	7	2 seconds	64 MB	Exams
F	8	1 second	64 MB	Fractions
G	9	4 seconds	96 MB	Gavril's Rap
H	10	2 seconds	64 MB	Highways
I	11	1 second	64 MB	Irina Tornado
J	12	4 seconds	64 MB	Jack-pot

Your solution must read the input data from the standard input and write the results to the standard output. Output to the standard error stream is prohibited.

Unless explicitly stated in the problem statements, all the input elements may be separated by an arbitrary number of whitespace characters. All the input data are correct and satisfy specifications given in the problem statement.

The output of your program must exactly satisfy the output specification in the problem statement.

General Sponsors



Problem A. Anti-factorial**Time limit:** 1 second**Memory limit:** 64 MB

For any given natural number n the product $1 \cdot 2 \cdot \dots \cdot n$ is called n *factorial* and denoted by $n!$.

Given the value of $n!$ for some integer $n \geq 1$, your task is to find n .

Input

The single line of the input data contains the value of $n!$. The number of decimal digits in $n!$ does not exceed 255. Also, the given number does not contain leading zeroes.

Output

Output the single line containing the requested number n .

Example

stdin	stdout
1	1
120	5

Problem B. Battery Upgrade

Time limit: 1 second

Memory limit: 64 MB

The Outer Ring Excavations (ORE) transgalactic corporation builds a new space station in the distant 4C-B7X star system. This station will become an outpost for new mining missions in the Outer Ring and Cloud Ring regions of the Galaxy, which are rich with the most rare and valuable Arkonor and Bistot ores.

The most important components of the space station are the Core Fusion Reactor (CFR) and the Core Capacitor Battery (CCB). The former produces energy, and the latter accumulates the energy for further use. The engineers of the ORE corporation have the components to build the CCB of the initial capacity of C energy units, which is fully charged by the CFR in T time units. Fortunately, the CCB has N upgrade slots, and up to N extension modules can be installed into these slots. There are two kinds of extension modules: Capacitor Booster (CB) and Reactor Control Unit (RCU). A Capacitor Booster increases the CCB capacity by dC energy units (the effect of several CBs is cumulative). A Reactor Control Unit increases the power output of the CFR thus decreasing the full charge time of the CCB by $dT\%$ (the effect of several RCUs is multiplicative).

In order to use the CCB as efficiently as possible, the ORE engineers want to install the extension modules to achieve the maximum CCB charge rate. The CCB charge rate is calculated as the energy capacity divided by the full charge time.

Input

The single line of the input contains four integers: the capacity of the CCB C ($0 < C \leq 1\,000\,000$), the number of available extension slots in the CFR N ($0 < N \leq 1\,000\,000\,000$), the battery capacity increase provided by one CB dC ($0 < dC \leq 1\,000\,000$), the full charge time decrease provided by one RCU (in per cents) dT ($0 < dT < 100$).

Output

Output two integers A and B , where A is the number of CB extension modules, and B is the number of RCU extension modules, that satisfy the ORE engineers' requirement. If several solutions are possible, output the one with the least value of A .

Example

stdin	stdout
100 10 15 10	3 7
1000 500 500 20	2 498

Problem C. Control Tower

Time limit: 1 second

Memory limit: 64 MB

The Air Congestion Monitoring company performs airspace monitoring and controlling. The company must ensure the safety of flight for civil and military flying objects. We assume that all objects fly at the same altitude but with different speed. The whole airspace is divided into areas of responsibility, each managed by a dispatcher. Each area of responsibility has a rectangular form. The task of the dispatcher is to prevent the objects from dangerous approach to each other in his area of responsibility. Thus the dispatcher must be able to calculate the minimal distance between two flying objects in his area of responsibility, given their initial positions and speeds. Your task is to write a program to solve this problem for dispatchers.

The dispatcher does not care about distance between flying objects when one or both of them are out of his area of responsibility, neither he cares about their movements preceding the initial moment.

Input

The first four integers in the input stream determine the responsibility area of the dispatcher. It is a rectangle parallel to axis OX and OY and is described by the coordinates of the opposite corners x_1, y_1, x_2, y_2 . The next integer N ($2 \leq N \leq 500$) determines the number of flying objects. The following N quadruples of integers x_i, y_i, v_i, w_i determine the initial position of each flying object and its speed (v_i is the projection of the speed onto the OX axis, w_i is the projection of the speed onto the OY axis). The initial position may lie out of the area of responsibility. All the coordinates and speed projections are in the range $[-2^{31}, 2^{31} - 1]$. The area of the area of responsibility is greater than 0. The border of the rectangle belongs to the area of responsibility. Flying objects are material points.

Output

Output the minimal distance between all pairs of flying objects. The distance must be printed with maximal possible precision. If such a minimal distance does not exist, print the string “inf”. The minimal distance between a pair of flying objects does not exist, if there exists no moment of time, when the both flying objects are inside the area of responsibility.

Example

stdin	stdout
-100 -100 100 100 2 0 0 0 0 1 1 1 1	1.414213562373095
-100 -100 100 100 2 1 1 1 1 1000 1 0 0	inf

Problem D. Dinner Time

Time limit: 1 second

Memory limit: 64 MB

Winnie-the-Pooh was walking through the forest one day. He was humming to himself, and walking gaily along, when suddenly he came to a sandy bank, and in the bank was a large hole.

“Aha” said Pooh. “If I know anything about anything, that hole means Rabbit,” he said, “and Rabbit means Company,” he said, “and Company means Food and Listening-to-Me-Humming and such like.”

So he bent down, put his head into the hole, and called out: “Is anybody at home?”

“No!” said a voice from inside the hole; and then added, “You needn’t shout so loud. I heard you quite well.”

“Isn’t there anybody here at all?”

“Nobody.”

“Hallo, Rabbit, isn’t that you?”

“No,” said Rabbit.

“Well, could you very kindly tell me where Rabbit is?”

“He has gone to see his friend Pooh bear, who is a great friend of his.”

“But this is Me, Pooh bear.”

“Oh, well, then, come in.”

So Pooh pushed his way through the hole, and he got in. He noticed that the time was HH:MM:SSZZ.

“What about mouthful of something?” asked Rabbit. Pooh always liked a little something at this time and he was very glad to see Rabbit getting out the plates and mugs; and when Rabbit said, “Honey or condensed milk with your bread?” he was so excited that he said: “Both,” and then, so as not to seem greedy, he added, “But don’t bother about the bread, please.” And for a long time after that he said nothing ... until at last, humming to himself, he got up, shook Rabbit lovingly by the paw, and said that he must be going on.

“Must you?” said Rabbit politely. “Well good-bye, if you’re sure you won’t any more.”

“Is there any more?” asked Pooh quickly.

Rabbit took the covers off the dishes, and said, “No, there wasn’t.”

“I, thought not,” said Pooh, nodding himself. “Well good-bye, I must be going on.”

So he started to climb out of the hole. He pulled with his front paws, and pushed with his back paws, and in a little while his nose was out in the open again ... and then his ears ... and then his shoulders ... and then —

“Oh, help!” said Pooh. “I’d better go back.”

“I can’t do either!” said Pooh. “Oh, help and bother!”

“The fact is,” said Rabbit, “you’re stuck.”

“It all comes,” said Pooh crossly, “of not having front doors big enough.”

“It all comes,” said Rabbit sternly, “of eating too much. I thought one of us was eating too much,” said Rabbit, “and I knew it wasn’t me.” “Then there’s only one thing to be done,” he said. “We shall have to wait for you get thin again.”

“How long does getting thin take?” asked Pooh anxiously.

“Not more than 24 hours, I should think.”

“But I can’t stay here for a day and night!”

“You can stay here is all right. It’s getting you out which is so difficult. I’ll read you,” said Rabbit cheerfully.

“What about meals?” asked Pooh.

“I’m afraid no meals,” said rabbit.

Some time later Pooh felt himself getting slenderer. And he said, “Now!”

Rabbit took hold of Pooh’s front paws, and all Rabbit’s friends-and-relations took hold of Rabbit, and they all pulled together.... And then, Pooh said “Pop!” just as a cork were coming out of a bottle. And Rabbit and his friends-and-relations went head-over-heels backwards... and on the top of them came Winnie-the-Pooh free!

“How long I was there?” asked Pooh.

“Now, it is HH:MM:SSZZ,” said Rabbit.

Silly Bear! Help him to answer the last question.

Input

The first line of input contains the description of time at which Pooh has entered the hole. The second line contains the description of time at which Pooh was set free. Both descriptions are given in HH:MM:SSZZ format. Here HH denotes hours, MM denotes minutes, SS denotes seconds, and ZZ is either “PM” or “AM” (without quotes). All numbers (HH, MM, and SS) are in two-digit format.

The conversion from 24-hour to AM/PM time system works as follows. Consider a time HH:MM:SS in 24-hour system. The first hour of the day (from 00:00:00 to 00:59:59 inclusively) is mapped to 12:MM:SSAM. The following 11 hours (from 01:00:00 to 11:59:59 inclusively) are mapped to HH:MM:SSAM. The first hour after noon (from 12:00:00 to 12:59:59 inclusively) is mapped to 12:MM:SSPM. And finally, the last 11 hours (from 13:00:00 to 23:59:59 inclusively) are mapped to hh:MM:SSPM where hh is HH minus 12.

For example, 00:00:00 maps to 12:00:00AM, 00:45:11 — to 12:45:11AM, 03:15:17 — to 03:15:17AM, 12:00:00 — to 12:00:00PM, and 19:43:15 — to 07:43:15PM.

Output

Output a single line containing the time Pooh stayed in the hole. The time must be not less than 1 sec and not greater than 24 hours. The time should be given in HH:MM:SS format, where HH is the number of hours, MM is the number of minutes, and SS is the number of seconds. If some of these numbers are less than 10, use leading zero for padding.

Example

stdin	stdout
12:01:13AM 07:43:12PM	19:41:59
10:15:48PM 02:13:12AM	03:57:24

Problem E. Exams

Time limit: 2 seconds

Memory limit: 64 MB

University of Lebon Igg is the World leader in the area of Probable Research. The university publishes the very influential journal “Annals of Probable Research” and employs such famous scientists, as Edward Cussler. The university’s fame attracts people wishing to study in it. The university accepts K students each year. This year there were N applications submitted.

The entrance test to the university consists of M exams. The maximal mark for each exam does not exceed 100. All marks above “2” are considered *satisfactory*. If an applicant receives unsatisfactory mark, he (or she) is not admitted to the further exams and loose possibility to enter the university this year.

The *score* of an applicant is the sum of all marks gained by the applicant. The best K (according to their score) applicants are admitted to the university.

The score is called *lowest passing score*, if all applicants who obtained this or bigger score and received no unsatisfactory mark are admitted. The score is called *semi-passing score*, if only a part of applicants who obtained this score and received no unsatisfactory mark is admitted.

You are given a database containing the information about marks obtained by all applicants. Your task is to write a program, which calculates the semi-passing score, or reports that no semi-passing score exists.

Input

The first line of the input contains the number of applicants N ($0 \leq N \leq 100000$), the number K of admitted applicants ($0 < K \leq 10000$), the number of exams M ($0 < M \leq 10$).

Each of the following N lines contains M non-negative integers — marks for all exams for the corresponding applicant. The mark is 0 if the applicant is not admitted to the exam due to an unsatisfactory mark for one of the previous exams.

Output

Output one integer — the semi-passing score, or 0, if the semi-passing score does not exist.

Example

stdin	stdout
5 3 4 3 3 3 5 7 5 2 0 4 4 3 3 4 5 6 7 5 4 3 3	14
4 3 2 30 30 40 40 50 50 60 60	0

Problem F. Fractions

Time limit: 1 second

Memory limit: 64 MB

Consider a decimal fraction of the following form:

$$\alpha = 0.1\dots$$

Here ellipsis (...) denotes a sequence containing from zero to seven decimal digits. Let K be the total number of decimal digits after the decimal point in the given fraction.

Your task is to find a proper fraction $\frac{p}{q}$ (p, q are integers, $0 < p < q$) with the least possible denominator q such that the first K digits after the decimal point in its decimal representation coincide with the corresponding digits of α .

Input

The input consists of a single line containing the given fraction α . The first character in the line is '0', then '.', then '1', then $K - 1$ decimal digits follow.

Output

Output a single line containing two integers p and q — the nominator and the denominator (respectively) of the requested fraction.

Example

stdin	stdout
0.1	1 6
0.19999999	4000000 20000001

Problem G. Gavrilă's Rap

Time limit: 4 seconds

Memory limit: 96 MB

A popular rap artist Nick&For La Peace is preparing the release of a new album called "Gavrilă's Rap". The essence of rap compositions is the rhythm, i.e. periodic alternation of stressed and unstressed syllables in the text. The trade mark of our rap artist is reading of the literature classic texts in the style of rap, but to do this, the artist has to know the rhythmic structure of the text.

A *rhythmic pattern* is a non-empty sequence of stressed and unstressed syllables. The text starts with the beginning of the rhythmic pattern and ends with the end of the rhythmic pattern. The rhythmic pattern must cover the text without self-impositions and gaps.

Write a program that determines the length of the shortest rhythmic pattern of the given text. To assist this task your program is given the dictionary, where the sequence of stressed and unstressed syllables is given for each word.

Input

The first line of the input data contains the number K ($0 < K \leq 1024$), which is the number of words in the dictionary. The next K lines of the input describe the dictionary as follows. The line begins with the word, which is a non-empty sequence of characters with ASCII codes from 33 to 255. The length of the word does not exceed 255. No space characters appear before the word. After the word, separated with one space character, a non-empty sequence of '0' and '1' characters follows. '0' corresponds to an unstressed syllable, and '1' corresponds to a stressed syllable. The length of the sequence does not exceed 255. Immediately after the sequence the end-of-line character with ASCII code 10 follows. The words in the dictionary are lexicographically sorted in the ascending order. Each word appears in the dictionary only once.

The text follows the dictionary in the input data. The text consists of words, separated with arbitrary number of whitespace characters. Whitespace characters may appear before the first word and after the last word. Whitespace character is a character with ASCII code in range from 1 to 32. Character with ASCII code 0 does not appear in the text. The text contains at least one word. The text ends with the 'END' keyword, which is not a part of the text and does not appear in the dictionary.

The total number of syllables in the text does not exceed 17 000 000. The total size of the input does not exceed 2 Mb.

Output

Output the length of the shortest rhythmic pattern of the text.

Example

stdin	stdout
5 gavrila 010 pochtalionom 10101 pochtu 10 raznosil 101 sluzhil 01 sluzhil gavrila pochtalionom gavrila pochtu raznosil END	2

Problem H. Highways

Time limit: 2 seconds

Memory limit: 64 MB

The Flatland Federation consists of N cities connected by M highways. Each highway is bidirectional and connects a pair of distinct cities. Some pairs of distinct cities may be connected by more than one highway.

As the maintenance cost of these highways became infeasible high, the Government has decided some of the highways to go private. A number of private companies agreed to participate in the tender. These companies have reached the mutual agreement, so that for every highway it is known by now which company is willing to obtain it.

The Government insists that the following additional requirements must be fulfilled:

1. To encourage competition and avoid monopolization, no company should be allowed to privatize more than one highway.
2. Since most of citizens do not like to think much as they drive, it is required that for any two distinct cities there should be at most one route connecting them and passing exclusively through the privatized highways.

Your task is to find which highways should go private (satisfying the above conditions) in order to maximize the total number of privatized highways.

Input

The first line of the input contains integers N and M ($1 \leq N \leq 100$, $0 \leq M \leq 5000$). The following M lines contain the descriptions of highways, one per line. The i -th line ($1 \leq i \leq M$) has the form $a_i b_i c_i$, where a_i and b_i are the cities which are connected by the highway i and c_i is the company which may privatize this highway ($1 \leq a_i \neq b_i \leq N$, $1 \leq c_i \leq 100$).

Output

On the first line output the maximum number K of highway which can be made private. On the second line output K integers — the numbers of these highways. Each privatized highway should be listed exactly once.

If there are several possible solutions you may output any of them.

Example

stdin	stdout
4 5 1 2 1 3 1 1 2 3 1 1 4 2 3 4 3	3 1 4 5
4 6 1 2 1 2 1 2 2 3 1 3 2 2 3 4 1 4 3 2	2 4 1

Problem I. Irina Tornado

Time limit: 1 second

Memory limit: 64 MB

The New Orel city is situated on an island in the Rybinsk reservoir. Long time ago it was situated on the river bank, but after filling the reservoir the city became entirely below the water-level. To avoid flooding of the city a system of dams was built around and inside the city, so the city territory became islands. All the dams (even those inside the city) are above water-level.

Unfortunately, the city is currently in the area of disaster of enormous Irina tornado. The tornado partially destroyed some dams and the city is being flooded. Your task is the following: given the map of dams and places of breakage, determine the area of the city which is under danger of flooding.

The map of dams is given on the coordinate grid with the OX axis directed from West to East and the OY axis directed from South to North. The city is located entirely to the North-East from the origin point of the map. All the dams are directed strictly parallel to the axis.

Input

The input begins with an integer N ($4 \leq N \leq 25\,000$), which determines the number of dams in the city. Then N quadruples x_1, y_1, x_2, y_2 follow, which determine the segments on the map corresponding to the dams ($0 < x_1, y_1, x_2, y_2 \leq 500$). No two dams have a common segment. The area of the city circled by the dams is greater than 0.

Then the integer M ($0 \leq M \leq 25\,000$) follows, which determines the number of destroyed parts of the dams. On the city map each destroyed part is represented by a segment of length 1 and is determined by a quadruple of integers — coordinates of the endpoints of the segment.

Output

Output the single integer — the area of the city part being flooded.

Example

stdin	stdout
4 2 2 4 2 2 2 2 4 4 2 4 4 2 4 4 4 1 2 2 2 3	4

Problem J. Jack-pot

Time limit: 4 seconds

Memory limit: 64 MB

A popular company named ACM (Alliance for Cheating and Misappropriation) conducts the following weekly lottery. During the week preceding the drawing participants make their stakes. A stake is an M -digit integer number with base K (i.e. a stake is a sequence of M digits, where each digit is from 0 to $K - 1$ inclusively). The leading zeros are allowed.

At the end of the week no more stakes are accepted and the company declares a winning number w (which is also an M -digit integer with base K). The participants who have guessed correctly the first digit of w (but haven't guess the second one) receive A_1 dollars each; those who have guessed correctly the first two digits of w (but haven't guess the third one) receive A_2 dollars each and so on. Finally, those who have guessed correctly all M digits receive A_M dollars each. (More formally: for any given stake let t denote the largest integer such that the first t digits of the stake coincide with the corresponding digits of w . If $t = 0$ no prize is paid, otherwise prize is A_t .)

Since the TV company knows all the stakes in advance, this makes cheating possible. And moreover, you are hired to do this job! Your task is the following: given the stakes made by the participants, you should choose w in order to minimize the total sum of prize money to be paid.

Input

The first line of input contains integers N (the number of stakes, $1 \leq N \leq 100\,000$), M (the length of each stake and the winning number, $1 \leq M \leq 10$), and K ($1 < K \leq 10$). The second line contains numbers $A_1, A_2, \dots, A_M$ ($1 \leq A_1 \leq A_2 \leq \dots \leq A_M \leq 1\,000\,000$). The following N lines contain stakes, one per line. Each stake is an M -digit number with base K . The stakes are given in non-decreasing order.

Output

In the first line of output print the winning number w which minimizes the total sum of prize money (if there are several numbers having the same sum the choice between them is arbitrary). In the second line of output print the total sum of prize money corresponding to w .

Example

stdin	stdout
10 3 2 1 3 100 000 000 001 010 100 100 100 100 110 111	011 6
1 1 10 100 0	7 0