NCPC 2015

October 10, 2015

ACM International Collegiate Programming Contest

IBM event sponsor

Problems

A Adjoin the Networks
B Bell Ringing
C Cryptographer's Conundrum
D Disastrous Downtime
E Entertainment Box
F Floppy Music
G Goblin Garden Guards
H Hero Power
I iCar
J Just a Quiz

Do not open before the contest has started.

Advice, hints, and general information

- Your submissions will be run multiple times, on several different input files. If your submission is incorrect, the error message you get will be the error exhibited on the first input file on which you failed. E.g., if your instance is prone to crash but also incorrect, your submission may be judged as either "wrong answer" or "run time error", depending on which is discovered first.
- For problems with floating point output, we only require that your output is correct up to either a relative or absolute error of 10^{-6}. For example, this means that
  - If the correct answer is 0.05, any answer between 0.049999 and $.050001$ will be accepted.
  - If the correct answer is 50, any answer between 49.99995 and 50.00005 will be accepted.

  Any reasonable format for floating point numbers is acceptable. For instance, "17.000000", "0.17e2", and "17" are all acceptable ways of formatting the number 17. For the definition of reasonable, please use your common sense.

Problem A
Adjoin the Networks
Problem ID: adjoin

Wikimedia, cc-by-sa

One day your boss explains to you that he has a bunch of computer networks that are currently unreachable from each other, and he asks you, the cable expert's assistant, to adjoin the networks to each other.

He has asked you to use as few cables as possible, but the length of the cables used does not matter to him, since the cables are optical and the connectors are the expensive parts. Your boss is rather picky on cable usage, so you know that the already existing networks have as few cables as possible.

Due to your humongous knowledge of computer networks, you are of course aware that the latency for an information packet travelling across the network is proportional to the number of hops the packet needs. And since you believe a good solution to your boss' problem may earn you that long wanted promotion, you decide to minimise the maximum number of hops needed between any pair of network nodes.

Input

On the first line, you are given two positive integers, the number $1 \leq c \leq 10^5$ of computers and the number $0 \leq \ell \leq c - 1$ of existing cables. Then follow ℓ lines, each line consisting of two integers a and b, the two computers the cables connect. You may assume that every computer has a unique name between 0 and $n - 1$.

Output

The maximum number of hops in the resulting network.

Sample Input 1	Sample Output 1
6 4 0 1 0 2 3 4 3 5	3

Sample Input 2	Sample Output 2

<pre>11 9
0 1
0 3
0 4
1 2
5 4
6 4
7 8
7 9
7 10</pre>
	<pre>4</pre>

Problem B
Bell Ringing
Problem ID: bells

Method ringing is used to ring bells in churches, particularly in England. Suppose there are 6 bells that have 6 different pitches. We assign the number 1 to the bell highest in pitch, 2 to the second highest, and so on. When the 6 bells are rung in some order—each of them exactly once—it is called a *row*. For example, $1, 2, 3, 4, 5, 6$ and $6, 3, 2, 4, 1, 5$ are two different rows.

An *ideal* performance contains all possible rows, each played exactly once. Unfortunately, the laws of physics place a limitation on any two consecutive rows; when a bell is rung, it has considerable inertia and the ringer has only a limited ability to accelerate or retard its cycle. Therefore, the position of each bell can change by at most one between two consecutive rows.

The pattern for an ideal performce, going through all rows, is depicted in Figure 2.

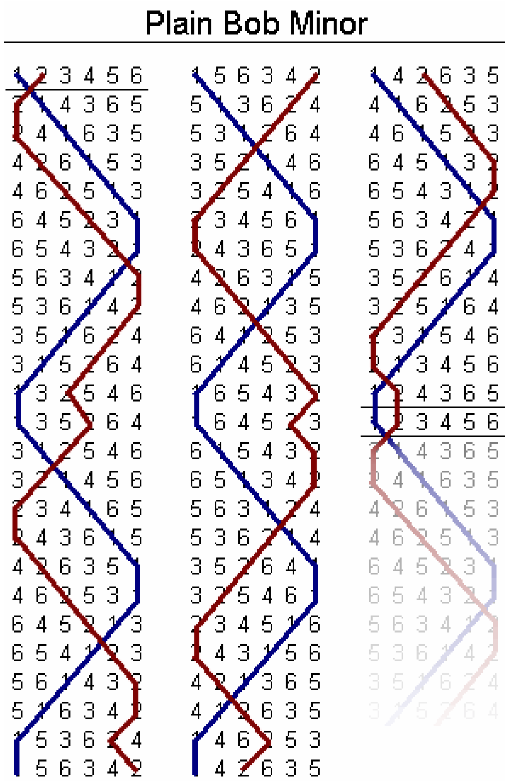


Figure B.1: It takes $6! = 120$ ringings to give all possible rows. The trajectory of bell number 1 is marked with a blue line and trajectory of bell number 2 marked with a brown line.

Given n, the number of bells, output an ideal performance. All possible rows must be present exactly once, and the first row should be $1, 2, \ldots, n$.

Input

The first and only line of input contains an integer n such that $1 \leq n \leq 8$.

Output

Output an ideal sequence of rows, each on a separate line. The first line should contain the row $1, 2, \ldots, n$ and each two consecutive lines should be at most 1 step away from each other. Each row should occur exactly once in the output.

Sample Input 1	Sample Output 1
2	1 2 2 1

Problem C
Cryptographer's Conundrum
Problem ID: conundrum

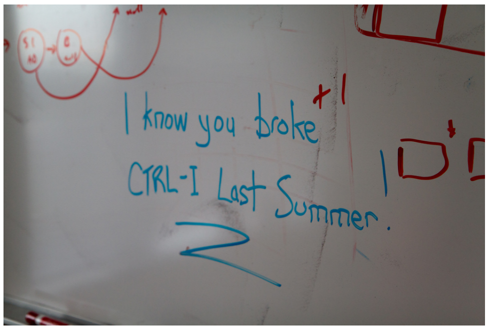


Photo by Alan Wu

The walls of the corridors at the Theoretical Computer Science group (TCS) at KTH are all but covered with whiteboards. Some of the faculty members are cryptographers, and like to write cryptographic puzzles on the whiteboards. A new puzzle is added whenever someone discovers a solution to the previous one.

When Per walked in the corridor two weeks ago, he saw that the newest puzzle read "`GuvfVfNGrfg`". After arriving at his computer, he quickly figured out that this was a simple ROT13 encryption of "`ThisIsATest`".

The series of lousy puzzles continued next week, when a new puzzle read "`VmkgdGFyIHPDpGtlcmhldGVuIHDDpSBzdMO2cnN0YSBhbGx2YXIK`". This was just base64-encoded text! "Enough with these pranks", Per thought; "I'm going to show you!"

Now Per has come up with a secret plan: every day he will erase one letter of the cipher text and replace it with a different letter, so that, in the end, the whole text reads "`PerPerPerPerPerPerPer`". Since Per will change one letter each day, he hopes that people will not notice.

Per would like to know how many days it will take to transform a given cipher text into a text only containing his name, assuming he substitutes one letter each day. You may assume that the length of the original cipher text is a multiple of 3.

For simplicity, you can ignore the case of the letters, and instead assume that all letters are upper-case.

Input

The first and only line of input contains the cipher text on the whiteboard. It consists of at most 300 upper-case characters, and its length is a multiple of 3.

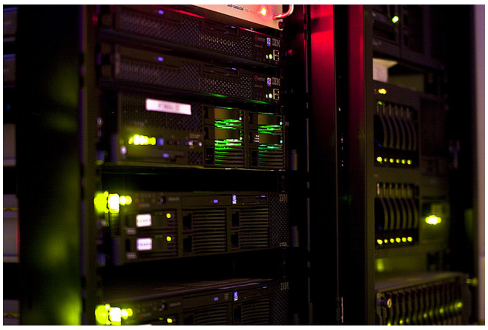
Output

Output the number of days needed to change the cipher text to a string containing only Per's name.

Sample Input 1	Sample Output 1
`SECRET`	`4`

This page is intentionally left (almost) blank.

Problem D
Disastrous Downtime
Problem ID: downtime



Claus Rebler, cc-by-sa

You're investigating what happened when one of your computer systems recently broke down. So far you've concluded that the system was overloaded; it looks like it couldn't handle the hailstorm of incoming requests. Since the incident, you have had ample opportunity to add more servers to your system, which would make it capable of handling more concurrent requests. However, you've simply been too lazy to do it—until now. Indeed, you shall add all the necessary servers …very soon!

To predict future requests to your system, you've reached out to the customers of your service, asking them for details on how they will use it in the near future. The response has been pretty impressive; your customers have sent you a list of the exact timestamp of every request they will ever make!

You have produced a list of all the n upcoming requests specified in milliseconds. Whenever a request comes in, it will immediately be sent to one of your servers. A request will take exactly 1000 milliseconds to process, and it must be processed right away.

Each of your servers can process at most k requests at once. Given this limitation, can you calculate the minimum number of servers needed to prevent another system breakdown?

Input

The first line contains two integers $1 \leq n \leq 100\,000$ and $1 \leq k \leq 100\,000$, the number of upcoming requests and the maximum number of requests per second that each server can handle.

Then follow n lines with one integer $0 \leq t_i \leq 100\,000$ each, specifying that the ith request will happen t_i milliseconds from now. The timestamps are sorted in chronological order. It is possible that several requests come in at the same time.

Output

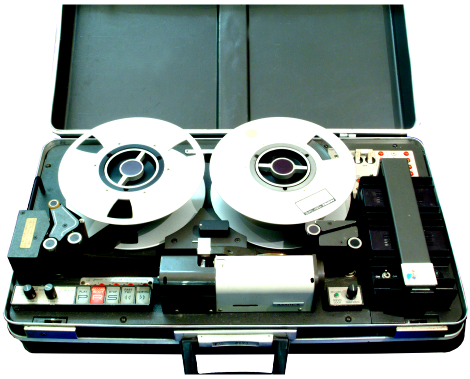
Output a single integer on a single line: the minimum number of servers required to process all the incoming requests, without another system breakdown.

Sample Input 1	Sample Output 1
2 1 0 1000	1

Sample Input 2	Sample Output 2
2 1 0 999	2

This page is intentionally left (almost) blank.

Problem E
Entertainment Box
Problem ID: entertainmentbox



Wikimedia, cc-by-sa

Ada, Bertrand and Charles often argue over which TV shows to watch, and to avoid some of their fights they have finally decided to buy a video tape recorder. This fabulous, new device can record k different TV shows simultaneously, and whenever a show recorded in one the machine's k slots ends, the machine is immediately ready to record another show in the same slot.

The three friends wonder how many TV shows they can record during one day. They provide you with the TV guide for today's shows, and tell you the number of shows the machine can record simultaneously. How many shows can they record, using their recording machine? Count only shows that are recorded in their entirety.

Input

The first line of input contains two integers n, k ($1 \leq k < n \leq 100\,000$). Then follow n lines, each containing two integers x_i, y_i, meaning that show i starts at time x_i and finishes by time y_i. This means that two shows i and j, where $y_i = x_j$, can be recorded, without conflict, in the same recording slot. You may assume that $0 \leq x_i < y_i \leq 1\,000\,000\,000$.

Output

The output should contain exactly one line with a single integer: the maximum number of full shows from the TV guide that can be recorded with the tape recorder.

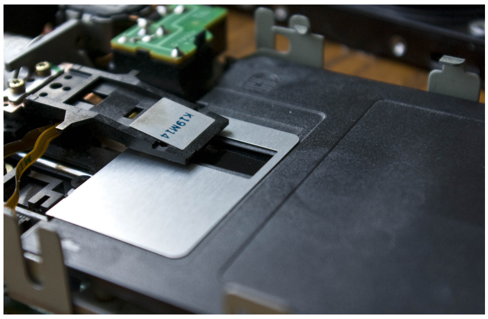
Sample Input 1	Sample Output 1
3 1 1 2 2 3 2 3	2

Sample Input 2	Sample Output 2
4 1 1 3 4 6 7 8 2 5	3

Sample Input 3	Sample Output 3
5 2 1 4 5 9 2 7 3 8 6 10	3

NCPC 2015

Problem F
Floppy Music
Problem ID: floppy



Antoine Taveneaux, cc-by-sa

Your friend's newest hobby is to play movie theme songs on her freshly acquired floppy drive organ. This organ is a collection of good old floppy drives, where each drive has been tampered with to produce sound of a unique frequency. The sound is produced by a step motor that moves the read/write head of the floppy drive. The motion of the read/write head occurs along the radial axis of the drive's spin disk, and the sound from one drive will play continuously as long as the read/write head keeps moving in one direction; when the head changes direction, there is a brief pause of 1fs—one floppysecond, or about 100 microseconds. The read/write head must change direction when it reaches either the inner or the outer end point of the radial axis, but it can also change direction at any other point along this axis, as determined by your friend. The starting position of the read-write head can be chosen freely.

Your friend is a nutcase perfectionist, and will not accept any pauses where there are not supposed to be any; nor will she accept sound when there is meant to be silence. To figure out whether a given piece of music can be played—perfectly—on her organ, she has asked for your help.

For each frequency, you are given a list of intervals, each describing when that particular frequency should play, and you must decide if all of the frequencies can be played as intended. You can assume your friend has enough drives to cover all the required frequencies.

Input

The first line contains an integer f, $1 \leq f \leq 10$, denoting the number of frequencies used. Then follow f blocks, on the format:

- A single line with two integers t_i, $1 \leq t_i \leq 10\,000$ and n_i, $1 \leq n_i \leq 100$; the number of floppyseconds it takes for the read/write head of frequency i to move between the end points of its radial axis, and the number of intervals for which frequency i should play.
- n_i lines, where the j-th line has two integers $t_{i,2j}, t_{i,2j+1}$, where $0 \leq t_{i,2j}, t_{i,2j+1} \leq 1\,000\,000$, indicating that the i-th frequency should start playing at time $t_{i,2j}$ and stop playing at time $t_{i,2j+1}$. You can assume that these numbers are in stricly ascending order, i.e. $t_{i,1} < t_{i,2} < \cdots < t_{i,2n_i}$.

Output

If it is possible to play all the f frequencies as intended, output "`possible`". Otherwise output "`impossible`".

Sample Input 1	Sample Output 1

<pre>1
6 2
0 4
6 12</pre>
	`possible`

Sample Input 2	Sample Output 2

<pre>1
6 3
0 5
6 8
9 14</pre>
	`impossible`

Problem G
Goblin Garden Guards
Problem ID: goblingardenguards

Felipe Escobar Bravo, cc-by-nc-nd

In an unprecedented turn of events, goblins recently launched an invasion against the Nedewsian city of Mlohkcots. Goblins—small, green critters—love nothing more than to introduce additional entropy into the calm and ordered lives of ordinary people. They fear little, but one of the few things they fear is water.

The goblin invasion has now reached the royal gardens, where the goblins are busy stealing fruit, going for joyrides on the lawnmower and carving the trees into obscene shapes, and King Lrac Fatsug has decreed that this nonsense stop immediately!

Thankfully, the garden is equipped with an automated sprinkler system. Enabling the sprinklers will soak all goblins within range, forcing them to run home and dry themselves.

Serving in the royal garden guards, you have been asked to calculate how many goblins will remain in the royal garden after the sprinklers have been turned on, so that the royal gardeners can plan their next move.

Input

The input starts with one integer $1 \leq g \leq 100\,000$, the number of goblins in the royal gardens.

Then, for each goblin follows the position of the goblin as two integers, $0 \leq x_i \leq 10\,000$ and $0 \leq y_i \leq 10\,000$. The garden is flat, square and all distances are in meters. Due to quantum interference, several goblins can occupy exactly the same spot in the garden.

Then follows one integer $1 \leq m \leq 20\,000$, the number of sprinklers in the garden.

Finally, for each sprinkler follows the location of the sprinkler as two integers $0 \leq x_i \leq 10\,000$ and $0 \leq y_i \leq 10\,000$, and the integer radius $1 \leq r \leq 100$ of the area it covers, meaning that any goblin at a Euclidean distance of at most r from the point (x_i, y_i) will be soaked by this sprinkler. There can be several sprinklers in the same location.

Output

Output the number of goblins remaining in the garden after the sprinklers have been turned on.

Sample Input 1	Sample Output 1
5 0 0 100 0 0 100 100 100 50 50 1 0 0 50	4

Problem H
Hero Power
Problem ID: heropower

Photo by friskytuna, cc-by-sa

Rhythm gaming seems to be having a bit of a renaissance this October, with both a new "Rock Band" and a "Guitar Hero" game coming out. Bj0rn is preparing to achieve top scores in "Guitar Hero Live", but he needs your help in figuring out what the maximum score is for all the new songs. Apparently, the new game has something called Hero Power, but Bj0rn is betting that it really is the same thing as the "Star Power" that has always been in these games.

"Guitar Hero's" scoring essentially works as follows: the player advances along a note chart and scores one point for each note he hits. Bj0rn will settle for nothing less than perfection; every note will be hit!

However, there's an added twist: "Star Power!"—simply called *SP*. Every now and then, a streak of star-shaped notes appear on the note chart. These streaks are *SP phrases*. When between the first and last note of an SP phrase, the player has the ability to charge up a so-called *SP meter*, which stores the amount of time the player has spent charging it. You can start charging at the exact moment of the first note and all the way up till the last note. You can also pause charging at any time and you do not have to use the accumulated SP immediately after you stop charging, so it is possible to accumulate SP charge from multiple phrases.

When the SP meter contains a positive amount of seconds, at any point in the song—even at the exact moment of a note—the player is free to *activate* Star Power. From this moment, the SP meter starts draining until it is completely empty. For example, if it contains $\pi + \sqrt[4]{7}$ seconds of SP, it will take $\pi + \sqrt[4]{7}$ seconds to drain completely. During an activation, every note is worth two points as long as the SP meter is non-empty! In particular, if you start activating at the exact moment of a note, that note is already worth two points and if you hit a note during the last moment of activation, that note is only worth one point, because the SP meter has just become empty.

There is a downside to activating Star Power. If an SP activation overlaps with an SP phrase and the SP meter is positive at some point during the overlap, the SP phrase degrades back to plain notes. In particular, if you hit the first note of an SP phrase on the exact moment when the SP meter drains to 0, the SP phrase is not degraded. It's fine to activate mid-phrase, but the rest of the phrase still suffers from the overlap and disappears, so you can not charge more Star Power from that phrase.

Can you help Bj0rn find the best strategy and figure out how many points he can get?

Input

The first line of input consists of two integers $1 \le n \le 50\,000$ and $0 \le p \le 100$, the number of notes and SP phrases respectively. The second line is a strictly increasing sequence of n integers $0 \le t_i \le 50\,000\,000$, the positions of all notes in milliseconds. Then follow p lines containing two integers each, $0 \le s_i < e_i \le 50\,000\,000$, the positions of the start and end of the i'th Star Power phrase.

Notes are guaranteed to exist on the start and end positions of each SP phrase. SP phrases never overlap and are given in ascending order.

Output

The maximum score as a single integer.

Sample Input 1

```
3 1
0 10 20
0 10
```

Sample Output 1

```
4
```

Sample Input 2

```
6 1
0 10 20 26 40 50
0 40
```

Sample Output 2

```
9
```

Sample Input 3

```
10 2
0 10 20 30 40 50 60 70 80 90
0 40
70 80
```

Sample Output 3

```
14
```

Problem I
iCar
Problem ID: icar

Cropped from picture by Les Chatfield

You are at home and about to drive to work. The road you will take is a straight line with no speed limit. There are, however, traffic lights precisely every kilometer, and you can not pass a red light. The lights change instantaneously between green and red, and you can pass a light whenever it is green. You can also pass through a light at the exact moment of changing colour. There are no traffic lights at the start or the end of the road.

Now your car is special; it is an iCar, the first Orange car, and it has only one button. When you hold down the button, the car accelerates at a constant rate of $1\mathrm{m}/\mathrm{s}^2$; when you release the button the car stops on the spot.

You have driven to work many times, so you happen to know the schedules of the traffic lights. Now the question is, how quickly can you get to work?

Input

The first line contains a single integer n, the length of the road in kilometers ($1 \leq n \leq 16$). Each of the next $n-1$ lines contains 3 integers t_i, g_i and r_i, the first time the i-th light will switch from red to green after this moment in time, the green light duration, and the red light duration ($40 \leq g_i, r_i \leq 50$; $0 \leq t_i < g_i + r_i$). Times are given in seconds. You may assume that any light with $t_i > r_i$ is currently green, and switches to red at time $t_i - r_i$.

Output

Output the minimum time required to reach the end of the road. Answers within a relative or absolute error of 10^{-6} will be accepted.

Sample Input 1	Sample Output 1
1	44.72135955

Sample Input 2	Sample Output 2
2 50 45 45	68.52419365

Sample Input 3	Sample Output 3
2 25 45 45	63.2455532

This page is intentionally left (almost) blank.

Problem J
Just a Quiz
Problem ID: justaquiz


Vincents Stuhl mit Pfeife

In the TV quiz *Monstermind*, a contestant chooses a topic and is then asked questions about it during a fixed period of time. The contestant earns one point for each correct answer. When the time runs out, the contestant must be silent.

Teresa has figured out such a niche topic that she knows all possible questions that may be asked about it, as well as all the answers. Since the competition is fierce, she has decided to sometimes answer a question before the host finishes reading it. The host picks each question uniformly at random from the pool of possible questions, and each question may be asked multiple times. When reading a question, the host reads at a pace of one word per second.

Teresa can interrupt the host mid-question—between words, or even before hearing the first word—but not mid-word—that would be extremely impolite. Answering also takes one second, and the host will start reading another question immediately after an answer—unless Teresa interrupts again.

She wrote a program to help her choose the best moment to answer, and now there is only one question left for you. How many points does she expect to score?

For example, in the first sample test case the answer is completely determined after hearing one word, so it is optimal to answer after hearing it, and Teresa answers 2 questions correctly in 4 seconds. In the second sample test case, if the first word is `What`, then it takes too much time to wait for the question to finish. Therefore Teresa says `Now!` 4 times and expects to get $1/3$ of the answers right.

Input

The first line contains two integers t and n ($1 \leq t \leq 100$, $1 \leq n \leq 100\,000$), the duration of the quiz and the number of questions. Each of the following n lines contains a question, which is a space-separated list of words terminated by a question mark; and an answer, which is a single word.

Each word is a sequence of non-space ASCII printable characters, between the ASCII values of ‘`!`’ and ‘$\sim$’. Only the last word of a question has a question mark (‘`?`’). You can assume that no question is a prefix of another and that punctuation marks are part of a word. Words spelled with different upper/lower case are assumed to be different.

It is guaranteed that the total number of word characters is at most $100\,000$.

Output

Output the expected score of an optimal strategy. Answers within a relative or absolute error of 10^{-6} will be accepted.

Sample Input 1

```
4 4
How much is 6 times 9? 42
How much is 9 times 6? 42
Is there intelligent life on Earth? Probably
What is the air speed velocity of an unladen swallow? African?
```

Sample Output 1

```
2.0000000000
```

Sample Input 2

```
4 3
What do we send? Code
What do we want? Accepted
When do we want it? Now!
```

Sample Output 2

```
1.333333333
```