Armenian Open Programming Contest

In memory of Vladimir Yeghiazaryan

May 24, 2020

Problems[1]

A	Sanitising	stdin/stdout
B	Hidden word	output-only
C	Drinking party	interactive
D	Look-and-Say numbers	stdin/stdout
E	Roman road	stdin/stdout
F	Formula One	output-only
G	Can you unzip me?	output-only
H	Virus shapes	output-only
I	Earthovirus	interactive
J	Statistics	stdin/stdout
K	Modelling viruses	stdin/stdout
L	Contest	stdin/stdout
M	Buildings	stdin/stdout
N	Lockdown	stdin/stdout

The contest was prepared by
Eduard Piliposyan and Hayk Saribekyan

We thank our partners

ՁԵՌՆԱՐԿՈՒԹՅՈՒՆՆԵՐԻ ԻՆԿՈՒԲԱՏՈՐ ՀԻՄՆԱԴՐԱՄ
ENTERPRISE INCUBATOR FOUNDATION

ONEMARKETDATA

РОССИЙСКО-АРМЯНСКИЙ УНИВЕРСИТЕТ

[1] See the last page for general information about the problems.

Problem A
Sanitising

Germs are everywhere! They can get onto hands and items we touch during daily activities and make you ill. Cleaning hands with soap and water or hand sanitizer is one of the most important steps you can take to avoid getting sick and spreading germs. If soap and water are not available, you should use an alcohol-based hand sanitizer that contains at least 60% alcohol.

During COVID-19 the above statement has been the top recommendation to protect against the spread of the pandemic. As a socially responsible person Eduard decided to follow the recommendations and buy a sanitizer. But the nearest pharmacy did not have a sanitizer with 60% alcohol content available, so Eduard decided to buy X liters of 96% alcohol and prepare a sanitizer with 60% alcohol content by mixing water (0% alcohol) with it.

He needs your help to calculate how much water to mix with X liters of **96%** alcohol to get his home-made sanitizer with **60%** alcohol content.

Input

The only line of the input contains one integer $X (1 \leq X \leq 100)$.

Output

The only line of the output must contain one number Y the amount of water that Eduard needs to mix, in liters. Print the answer with 2 digits after the decimal point.

Sample Input 1	Sample Output 1
`10`	`6.00`

Sample Input 2	Sample Output 2
`15`	`9.00`

Problem B
Hidden word

It's spring of 2020. The active life of Yerevan has stopped. Everyone is bored, especially Hayk and Bell, so they decided to play a game.

The game is simple: Bell thinks of a word W, and Hayk has to guess W. To give Hayk some chance, Bell has given Hayk a big table of letters. As a hint, for each letter of W, Bell tells Hayk a location in the table where that letter appears. The location is given as a (row, column) pair, with indexing starting from 1.

For example, if Bell's hidden word is "dur", he could give Hayk the following locations:

```
(3,3), (4,4), (2,3).
```

After winning 2491 rounds of this game, Hayk needs your help to win the last 2492th round. The hint from Bell for this round is

```
(17,32), (23,50), (29,59), (35,64),
(42,77), (48,86), (56,92), (62,100).
```

Input

There is no input for this problem, except for the link above. Here is an alternative link to the page.

Output

You should simply upload or paste Bell's hidden word in the online judge, on a single line, without spaces. Below is a (wrong) answer in the correct formatting:

```
yerevan
```

Problem C
Drinking party

After a big party, there are n dirty glasses in the living room that your friend Alice and the Jury need to take to the kitchen. Since the glasses are fragile, both of them can only take one in each hand. So on each trip from the living room, they can take either one or two glasses. Also, the door is small so they alternate: first Alice takes some glasses, then the Jury, then Alice again and so on until they are done.

The person who takes the last glass to the kitchen will have to wash all the glasses. Clearly, neither Alice nor the Jury want to wash the glasses. The Jury has written a program to determine whether he should take one or two glasses at each point to avoid washing the glasses, if possible. Your task is to write a program to help Alice to determine her strategy for not taking the last glass.

The interaction between your program and Jury's program. This is an interactive problem, where your program will interact with the Jury's program according to the following protocol.

First, your program must read from the standard input the total number of glasses N (the Jury has counted them for you). After this, your program must write in the standard output the number of glasses Alice should take to the kitchen (either 1 or 2) when she makes the first trip. If only one glass is remaining, your program must finish as Alice won. Otherwise, your program must read the number of glasses the Jury takes on their turn (again, either 1 or 2). Then, the process repeats as it is again Alice's turn to take glasses.

When your program makes its last move or receives EOF (end-of-file) signal, it must quit its execution. Do not forget to flush after each output. See the last page for more details about interactive problems.

Limits. It is guaranteed that $1 \leq N \leq 10^4$. It is also guaranteed that n is such that Alice always has a chance to force the Jury to take the the last glass.

Note, in the examples below, the empty lines are only for presentation. Your program must not produce them.

Jury's output 1	Your output 1
11	
	1
1	
	2
2	
	1
2	
	1

Problem D
Look-and-Say numbers

The mathematician John H. Conway (26 December 1937 – 11 April 2020) was famous for many games he invented. One of those games is Look-and-Say, which is a one-player game that generates a sequence of numbers, starting from 1. Each subsequent number is constructed from the previous one using a simple look-and-say rule: Split the previous number into groups of consecutive equal digits, then read off the groups by saying the number of digits in each group and then the digit itself. It is easier to understand the rule by examples:

- 1 is read off as "one 1" or 11;
- 2223344 is read off as "three 2s, two 3s, two 4s" or 322324;
- 1111332 is read off as "four 1s, two 3s, one 2" or 412312.

Thus, the first numbers of the sequence are

$$1, 11, 21, 1211, 111221, \ldots$$

Your task is to find kth number in the sequence.

Input

The only line of the input contains one integer k $(1 \leq k \leq 16)$.

Output

The only line of the output must contain the kth number in the Look-and-Say sequence.

Sample Input 1	Sample Output 1
`1`	`1`

Sample Input 2	Sample Output 2
`4`	`1211`

Sample Input 3	Sample Output 3
`5`	`111221`

Problem E
Roman road

The towns of Cowford and Wentbridge are connected by an old one-way Roman road. The roads that Romans built are very straight, so we will represent the road between Cowford and Wentbridge as a straight line of length L meters.

The Roman roads were not designed for cars, so some cars cannot pass one another. In particular, if a faster car of width w_F approaches a slower car of width w_S, then the faster car can only overtake if $w_F + w_S \leq W$, where W is the width of the road. Otherwise, the faster car is stuck behind the slower one and the two cars have to drive at the same (slower) speed. Since the cars are not very long, we will assume that the cars in such a situation arrive to Wentbridge at the same time, i.e., we assume that the cars are points.

On the weekend, the N residents of Cowford go to Wentbridge in their cars. Resident i leaves Cowford at time t_i, and moves at speed s_i meters per second towards Wendbridge, along the Roman road. Additionally, the width of the car of resident i is w_i microns.

The residents have asked you to write a program that, for each resident, determines the time when they arrive to Wentbridge.

Input

The first line of the input contains three integers: N $(1 \leq N \leq 10^5)$, the number of Cowford residents; $L(1 \leq L \leq 10^9)$ the length of the Roman road in meters; $W(1 \leq W \leq 10^7)$ the width of the roads in microns.

Line i of the input after that contains three integers: $t_i(0 \leq t_i \leq 10^9)$, the time resident i left Cowford; $s_i(1 \leq s_i \leq 10^9)$, the speed at which that resident travels in meters per second; w_i $(1 \leq w_i \leq W)$, the width of the ith resident's car in microns.

If two cars leave Cowford at the same time, then the one with a smaller index always goes first.

Output

The ith line of the output contains on real number – the time, in seconds, when the ith resident of Cowford arrives to Wentbridge. Print the answers with at least 3 digits after the decimal point.

Sample Input 1	Sample Output 1
1 1 1 1 1 1	2.00000

Sample Input 2	Sample Output 2
4 10 3 7 8 3 3 3 2 1 1 1 5 5 2	11.00000 6.33333 11.00000 7.00000

Problem F
Formula One

As we already know, Eduard likes to look at and collect statistics very much. And he is also a Formula One fan. He decided to make his personal list of F1 results from different years and races, to show to his friends. Eduard only cares about The Grands Prix, which is a series of races that take place across the world throughout the year.

He prepared a list of questions about the information he is missing in one file: formula1.txt. Help him to find the answers. Eduard will be happy if you make at most 15 mistakes because in that case none of his friends will notice.

Input

There is no input for this problem, except for the link above. Here is an alternative link to the page.

Output

You should simply upload or paste list of companies that took the corresponding place on the given round of the give year. Your output must have 463 lines.

Below are the first 5 lines of the output in the correct format:

```
Ferrari
Arrows
McLaren
Mercedes
Tyrrell
```

Your solution will be accepted if you have 15 or less mistakes.

Problem G
Can you unzip me?

To spend the lockdown period effectively Eduard decided to clean his computer's hard disk from unnecessary files. He knew that there were many of them, but he never expected to find a file called "big.txt" of about 400 GB in size. When Eduard tried to open the file, his text editor crashed. He can't remember how and why this file was created.

Fortunately Eduard found another file called "small.txt", and he was able to open it using his text editor. This file was full of whitespace, but contained one hidden phrase somewhere in the middle of it. The hidden phrase was "happy_new_year".

Seeing this, Eduard remembered that "big.txt" has a similar format, i.e., it mostly contains a whitespace with exactly one non-whitespace phrase hidden in it.

He has asked you to find the hidden phrase in "big.txt". He zipped the file, and then zipped the resulting file again (double zipping), so he can give it to you. Thinking that it might help you, he has also double-zipped "small.txt" and given to you. Help him to find the phrase in "big.txt".

You can download both files by the following links:

small.zip
big.zip

Input

There is no input for this problem, except for the zip files above.

Output

You should simply upload or paste the hidden phrase of "big.txt" file in the online judge.

Problem H
Virus shapes

Your are an expert virologist and over the past decades have discovered many interesting facts about viruses. Your first discovery was that all the viruses look like a convex polygons. Later you discovered that the internal angles of all viruses (polygons) are between 45 and 170 degrees, because otherwise they cannot live.

Your new theory is that the number of corners of the virus (polygon) determines how dangerous the virus is. To test this hypothesis, you developed a microscope that can take images of any virus. The result is an image that shows only the "shell" or the "boundary" of the virus, which is enough to determine the number of its corners. Additionally, your genius microscope has such a high resolution that the sides of all the viruses (polygons) has at least 150 pixels in the image.

Your assistant has collected a database of images of 3000 viruses. To test your theory, your first task is to find the number of corners of each of the 3000 viruses. Correctly determining the number of corners on 99% of the viruses will be sufficient to test your theory and win a Nobel prize.

Input

There is no input for this problem, except for the link above. Here is an alternative link to the page.

Output

You should simply upload or paste a space-separated list containing the number of corners of each virus in the online judge, on a single line. For example, below is a (wrong) answer with in the correct format:

```
5 3 7
```

If your answer is correct on at least 99% of the viruses, your solution will pass.

Problem I
Earthovirus

It is year 2200. Not only we have survived but you are on Mars and your mission is to isolate Mars from the only dangerous virus left in the universe: the Earthovirus. For that you have to compare the DNA of the suspected virus against the DNA of Earthovirus.

DNAs of viruses can be represented as strings composed of letters A, G, C, T. The complete DNA of Earthovirus is given by a string E of length n. Due to the radiation, the DNAs of all other viruses in the universe are uniformly random strings of arbitrary length.[2] Thus, if the DNA of the suspected virus is represented by the string X, then either $X = E$ or X is a random string composed of characters A, G, C, T.

When the laboratory processes the suspected virus, its DNA X breaks in some places. The parts that have less than 100 characters are very small and get lost. So, only some disjoint substrings $Z_1, Z_2, \ldots, Z_k$ of test DNA X remain and each substring Z_i is at least 100 characters long. The laboratory guarantees that in total at least $n/2$ characters remain in those substrings.

You can give the laboratory a string S representing a DNA code, and they will quickly check whether S is a substring of some Z_i. You need to write a program that asks no more than such 100 questions and determines whether the test virus is an Earthovirus or no.

The interaction between your program and the laboratory. This is an interactive problem. First, your program will need to read from the standard input the string E. After this, until your program finds the answer, it needs to write in the standard output a substring S, to be checked by the laboratory. If S is a substring of Z_i for some $i = 1, \ldots, k$, then your program will get an answer "Yes" in the standard input, otherwise it will receive an answer "No". If at any point your program has determined the type of the virus, it has to print either "Earthovirus" or "Randovirus", depending on which it is. After this, your program must stop.

When your program determines the answer or receives EOF (end-of-file) signal, it must quit its execution. Do not forget to flush after each output. See the last page for more details about interactive problems.

Limits. It is guaranteed that $1 \le n \le 10^5$. You are not allowed to ask more than 100 questions, and the total number of characters in the DNA-strings that you ask must not exceed 10^6. All your questions must be strings representing a valid DNA.

In the samples below it is assumed that no substring of X is lost, no matter how small, i.e., $Z_1, \ldots, Z_k$ represent a *complete* partition of X. This is done to save space in the presentation. Recall, that in the actual execution, the strings Z_i have at least 100 characters each and do not necessarily cover all of X.

The samples are only for demonstration of the interaction of your program and the judge's, they are created manually.

Note, in the examples below, the empty lines are only for presentation. Your program must not produce them.

[2] Each character of the string is one of the four possibilities, with probability $1/4$, independently from all other characters.

Laboratory's feedback 1	**Your questions 1**
`AGATACATAGACA`	
	`TAGA`
`No`	
	`TAC`
`Yes`	
	`ACAGA`
`No`	
	`GATACAT`
`Yes`	
	`Earthovirus`

Explanation of the sample 1: In this case, X is equal to E and it is broken into substrings $Z_1 =$ `AGATACATA` and $Z_2 =$ `GACA`.

Laboratory's feedback 2	**Your questions 2**
`GACTCACTACGAAT`	
	`CACT`
`No`	
	`CT`
`Yes`	
	`ACT`
`No`	
	`GAAT`
`No`	
	`Randovirus`

Explanation of the sample 2: In this case, $X =$ `TGCTAAGCAGATCGA` and is different from E, it is split into $Z_1 =$ `TGCTA`, $Z_2 =$ `AGCAG` and $Z_3 =$ `ATCGA`.

Problem J
Statistics

Eduard loves to look at statistics and during the COVID-19 pandemic he monitors its development. It has been exactly N days since the first case of the virus appeared, and Eduard has collected the number of new cases per day in an array D of length N. He thinks that one of the most important metrics is the maximum number of daily new cases of the virus during last K days. Eduard asks you to write a program which will help him to create this very important metric in a new array M.

Input

The first line of the input contains two integers $N (1 \leq N \leq 10^5)$, the number of days from the first reported case of the virus, and $K (1 \leq K \leq N)$. Second line contains N space-separated non-negative integers, the elements of the array D. The maximum number of daily new cases is 10^9.

Output

The only line of the output must contain N numbers – values of array M.

Sample Input 1	Sample Output 1
`7 3` `7 5 80 40 3 2 97`	`7 7 80 80 80 40 97`

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

Problem K
Modelling viruses

You have been asked to write a program to model the spread of a virus in a population. As an excellent programmer, you have split the task into simple sub-tasks. In your current sub-task you model the movement of two people, Alice and Bob. Your goal is to determine if they violate the social-distancing rules, and if so, when.

Alice and Bob are represented as points moving in the two dimensional space. Alice starts her movement at point A at time 0 and moves with a constant velocity given by a vector v_A. You have the same information about Bob. Both Alice and Bob move for exactly T seconds, and then stop.

Alice and Bob violate the social-distancing rule if their distance is 2 metres or less. Write a program that determines the earliest time when they violate the rule, if they ever do.

Input

The first line of the input contains a single integer $T (1 \leq T \leq 10^5)$, the length of the time interval of interest. The second line contains the information about Alice, four space-separated integers x, y, v_x, v_y, which are all at most 10^5 in absolute value. The starting position of Alice is $A = (x, y)$ and her velocity is given by the vector $v_A = (v_x, v_y)$. The third line contains the information about Bob, in the same format and satisfying the same conditions. All the values are in standard units, i.e., metres and seconds.

Output

The only line of the input should have the first time, in seconds, when Alice and Bob are violating the social-distancing rule. If this never happens, simply output -1. Your output should have at least 5 digits after the decimal point.

Sample Input 1	Sample Output 1
5 2 0 0 1 0 7 0 -1	3.500000

Sample Input 2	Sample Output 2
2 0 0 1 1 8 8 -1 -1	-1

Sample Input 3	Sample Output 3
5 -3 -1 2 3 4 2 -1 2	1.800000

Problem L
Contest

Hayk and Eduard like preparing programming competitions together and are now working on the problem set for CEC – Coronavirus End Cup, the largest programming contest in history. They have exactly T days until the competition. Hayk has created N problems, and he needs to convince Eduard that his problems are good enough for CEC.

To finalise the problem set, Hayk and Eduard meet once a day until the day of the competition (T times in total). For each meeting, Hayk focuses only on one of his problems and tries to convince Eduard to include that problem in CEC. If Eduard sees problem i for the first time, he will agree to include it in CEC with probability p_i, and reject it with probability $1 - p_i$. If it is rejected, Hayk can suggest the same problem on another meeting later. However, Eduard does not like to repeat things, and if Hayk proposes problem i for the second time, Eduard will agree with him with probability only $f \cdot p_i$. In general, the probability that Eduard accepts problem i on its kth proposal is $f^{k-1} \cdot p_i$, i.e., it decreases by factor f every time problem i is proposed by Hayk. Notice that the probability that Eduard will agree decreases only for problem i, and does not change for the rest of the problems.

Not all problems are created equal. Hayk has assigned a score S_i to problem i, which determines how much he likes that problem. He wants to increase the total score S of his problems at CEC. Hayk knows Eduard very well and has calculated the values p_i and f exactly. Help Hayk to compute the largest expected value of S that he can have.

Input

The first line of the input contains the integer $N (0 \leq N \leq 10)$, the number of Hayk's problems; the integer $T (0 \leq T \leq 100)$, the number of days until CEC; and the real number $f (0 \leq f \leq 1)$, the factor by which Eduard's mood changes.

N lines follow. The ith line contains the values $p_i (0 \leq p_i \leq 1)$, the probability that Eduard agrees problem i the first time he sees it; and $S_i (0 \leq S_i \leq 10000)$, the score of problem i.

Output

The only line of the output must contain the largest expected value of S, with at least 5 digits after the decimal point.

Sample Input 1	Sample Output 1
1 3 0.8 0.6 100	87.187200

Sample Input 2	Sample Output 2
2 3 0.6 0.5 50 0.5 10	39.650000

Problem M
Buildings

There is a lot of construction going on in Yerevan. But it is hard to guess where the new building will be constructed. In order to find their locations Eduard did some research and found an interesting pattern.

Let's represent Yerevan as a two dimensional rectangular grid, where buildings can be constructed only at integer-coordinated locations. No two buildings can occupy the same location. The following is the pattern that Eduard found: If three buildings stand at the corners of some rectangle with sides parallel to the coordinate axes, then a building will be constructed at the fourth corner, if it is empty. The rule will then be repeated using the newly built buildings too, until no more buildings can be built.

When Eduard did his research, there were already K buildings standing. He now asks you to write a program which calculates the number of buildings that will eventually be standing in Yerevan.

Input

The first line contains an integer $K (0 \leq K \leq 2 \cdot 10^5)$. After that, for each $i = 1, \ldots, K$, the ith line contains the coordinates of the ith existing building, integers X_i and $Y_i (-10^9 \leq X_i, Y_i \leq 10^9)$, separated by a space.

Output

Output the number of buildings in Yerevan at the end of all construction.

Sample Input 1

```
3
1 1
1 2
2 2
```

Sample Output 1

```
4
```

Sample Input 2

```
5
0 0
1 0
0 1
1 2
2 1
```

Sample Output 2

```
9
```

Problem N
Lockdown

The city where Ashot lives has implemented extremely strict lockdown rules: No one is allowed to go outside for any reason. There are police officers at many places in the town to enforce the law. This is terrible news for Ashot, because he walks to his favourite park every day for inspiration to solve the latest programming problems. So he decided to go to the park without being seen by the police officers.

Ashot has created a map of the city, which is represented by a grid with R rows and C columns. On his way to the park, Ashot can move from his current location (a cell in the grid) to a horizontally or vertically neighbouring location in one minute. Alternatively, Ashot can choose to stay at his current location for one minute.

Ashot has discovered that there are P police officers patrolling in the town. Each officer is assigned a fixed path of locations. The officer starts from the beginning of his path, and every minute moves on the next location on the path. When he reaches the last location of the path, he turns around and follows the path back. The officer continues his movement like this indefinitely.

Some locations of the city have museums, which are closed, so neither Ashot, nor the police officers can go in them. A police officer can see Ashot only if they are on the same row or column and there is no museum between them. Help Ashot to find the least amount of time (in minutes) that he will need to reach the park without being seen by any police officer.

Input

The first line of the input contains two space-separated integers R and C ($1 < R, C \leq 60$), the number of rows and columns on Ashot's grid-map of the city. The second line contains the locations of Ashot's home and his favourite park, separated by a space. Each location consists of a `(row column)` pair, written using the parentheses, as shown in the sample input. The numbering of rows and columns starts from 1.

R lines follow, each containing exactly C characters. The character '`.`' corresponds to a location where both Ashot and the police officers can walk, and '`#`' corresponds to museum.

The following line contains a single integer P ($1 \leq P \leq 200$), the number of police officers in the town. Then P lines follow. The ith line represents the ith police officer. It starts with an integer k_i ($1 \leq k_i \leq 7$), the number of location on the ith officer's path. Then the k_i locations on this officer's path follow as a space-separated list, in the same format as before.

It is guaranteed that the path of each officer is valid: It stays within the city, does not go through a museum and always moves in one of the 4 directions from any location. It is also guaranteed that Ashot's home location and the park do not have a museum.

Output

The only line of the output must contain the least amount of time, in minutes, that will take Ashot to reach the park from his home. If he cannot reach the park, simply output `-1`.

Sample Input 1

```
5 6
(5 3) (3 6)
......
..##.#
#.....
#.##.#
#...##
1
6 (2 5) (3 5) (3 4) (3 3) (3 2) (2 2)
```

Sample Output 1

```
26
```

Note that Ashot is seen by a police officer the moment he arrives to the park on minute 26. This is ok, because just a brief moment in the park is enough for him to get inspired.

Sample Input 2

```
4 4
(1 3) (4 3)
....
##..
....
.#.#
1
2 (3 2) (3 1)
```

Sample Output 2

```
-1
```

General information about interactive and stdin/stdout problems

- Time limit for all interactive or stdin/stdout problems is 2 seconds and memory limit is 128MB.
- All input/output should be done from standard input and output. You can find examples of standard input/output in common languages here.

General remarks about interactive problems

1. You must print a new line after each interaction;
2. You must flush the output stream after each interaction:
   - In C or C++: `fflush(stdout);`
   - In Java: `System.out.flush();`
   - In Python: `sys.stdout.flush()`
   - In C#: `Console.Out.Flush();`
3. If your program receives EOF (end-of-file) condition on the standard input, it *must* exit immediately with exit code 0. Failure to comply with this requirement may result in Time Limit Exceeded error instead of other verdicts AC/WA/PE.
4. Typical issues with interactive problems are
   - Wrong Answer – usually means that your program followed the interaction protocol but the answer or the intermediate steps are wrong.
   - (Wall) Time Limit Exceeded – this means that due to your program's execution, the interaction is not progressing. This can happen
     - if your program is expecting an input from the jury's program by mistake. Most commonly this happens when the jury's program has detected an error and quit, but your program is waiting for an input. In this case, you will get the TLE verdict, instead of the actual WA/PE error. See point (3) above;
     - if your program has not provided the necessary output for the jury's program to respond. Most often this is because you have not flushed the output stream. See point (2) above.
   - Presentation Error – usually means that your program did not follow the interaction protocol correctly and the jury's interacting protocol is not able to test it.
   - Runtime error – usually a mistake in your program that makes your program crash during the execution.