Spotify Code Quest 2012

February 18, 2012

Problems

- A Toy Railway
- B Evenland
- C Streaming Statistics
- D Greedy Cows
- E Three-State Memory
- F Troll Hunt
- G Smooth Monkey
- H Collapse

(the problems are ***not*** sorted by difficulty)

This page is intentionally left blank.

Problem A
Toy Railway

Nora is playing with her toy railway. Her railroad network consists of a number of railroad switches connected by tracks. As the network starts to be rather complex, she needs some help with the traffic planning. Write a program that determines how each switch should be set in order for Nora's train to make the shortest possible loop and return to the starting position.

Each switch looks as in Figure A.1 and can be set to either of two states: B or C. If the trains enters at A it goes to B or C depending on the state of the switch. On the other hand, if the train enters at B or C, it always goes to A regardless of the state of the switch.

Photo by Robin Webster

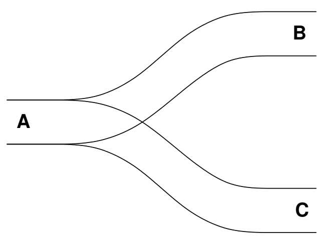

Figure A.1: Labelling of the three connection points of a switch.

The starting position of Nora's train is such that it is entering switch 1 at point A. The program should find static switch settings that make the train return to the starting position, faced in the same direction, while passing as few switches as possible, or determine that it is impossible. The train may travel on the same track multiple times (in different directions), but can never be reversed.

Input

The first line of the input contains two integers N and M, the number of switches and the number of connections, respectively, where $1 \leq N \leq 100\,000$ and $1 \leq M \leq 150\,000$. Then M rows follow, each containing two strings: the labels of two distinct connection points that are connected by track. Each label consists of an integer between 1 and N identifying the switch, and then a letter A, B or C, identifying the specific connection point, according to Figure A.1. All tracks can be traveled in both directions. No label occurs more than once and all labels do not have to occur (there may be dead ends and even disconnected parts of the network). A track may connect two points on the same switch. The network does not have to be realizable on a plane; Nora likes building in several levels.

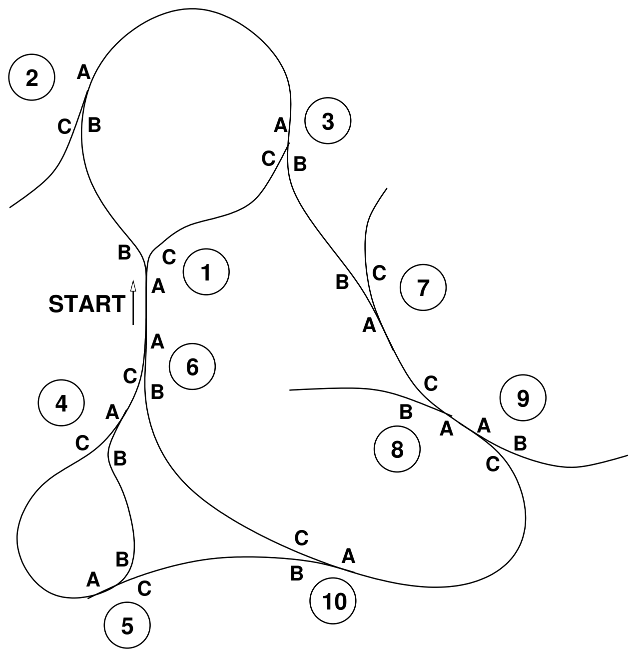

Figure A.2: The railway network in the first example below. The shortest loop contains 8 switches: $1 \to 2 \to 3 \to 7 \to 8 \to 9 \to 10 \to 6 \to 1$. There are several loops containing 9 switches, for example $1 \to 2 \to 3 \to 1 \to 6 \to 4 \to 5 \to 4 \to 6 \to 1$.

Output

If a solution exists, output a string of N characters, each being either B or C and giving the state of switch $1, 2, ..., N$, respectively. If there are multiple solutions, any one of them will be accepted. If no solution exists, output the string "`Impossible`".

Note that there are many possible solutions in the first example below: only the states of switches 1, 3, 9 and 10 matter.

Sample Input 1	Sample Output 1
10 13 1B 2B 1C 3C 3A 2A 4B 5B 4C 5A 1A 6A 6C 4A 3B 7B 7A 8C 8A 9A 9C 10A 10C 6B 10B 5C	BCBBBBBBCC

Sample Input 2	**Sample Output 2**

`2 3`
`1B 1C`
`1A 2C`
`2A 2B`
	`Impossible`

This page is intentionally left blank.

Problem B
Evenland

Evenland used to be a normal country. Then Steven became ruler and now everything must be done as he wishes. For some odd reason he is obsessed with the number two. Everything must be even in his country, hence he even changed the name to Evenland.

Picture by U.S. Fish and Wildlife Service in Public Domain

The other day, Steven was driving through his country when he noticed that, at some intersections, an odd number of roads meet. Naturally, some roads must now be destroyed in order to make the number of roads even at every intersection.

You are in charge of this project. You start wondering: in how many ways can this project be carried out? In other words, in how many ways can you select a set of roads to destroy so that all intersections become even? The resulting road network does not have to be connected, so for instance, one possible way is to destroy *all* roads.

Input

The first line of the input contains two integers N and M, where $1 \le N, M \le 100\,000$. N denotes the number of intersections in Evenland and M is the number of roads. M lines follow, each contains two space separated integers a, b indicating that there is a road between intersections a and b. You may assume that $1 \le a, b \le N$ and $a \neq b$. There might be more than one road connecting a pair of intersections.

Output

Output one line with one integer – the number of ways of making all intersections even. Since this number might be big, output the remainder modulo $1\,000\,000\,009$.

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

Sample Input 2	Sample Output 2
2 1 1 2	1

Problem C
Streaming Statistics

Photo by Joe Pritchard

Eddie River has been hired by a company providing a music streaming service. His boss, Veronica Brook, wants to get an analysis of network utilization. While there are a lot of different types of requests, the bulk of the data transfer is the streaming of music files.

When delivering streaming music one keeps track of which track was requested, when streaming completed (Unix epoch in milliseconds (ms)), its duration in milliseconds (ms), and its bitrate in kilobits per second (kbps). If a track has duration 100 000 ms and ended at 1325967178930, that means it started at 1325967078930 and played until 1325967178930. If its bitrate was 96 kbps then the amount ofa data streamed due to this song was 9 600 kb. During the interval between 1325967078930 and 1325967078931, 96 bits was streamed, and during the interval between 1325967178930 and 1325967178931 nothing was streamed (as it stopped at 1325967178930). Veronica tells Eddie he can make the assumption that start and stop times are exact, and also that at bitrate r kbps, exactly r bits are sent every millisecond (in reality data would be sent in "bursts" as larger packages).

Eddie's task is to analyze a simplified log file. The log file is a sequence of lines, each of which has three numbers, t, d, r which are the end time (Unix epoch in ms), duration (ms) and bitrate (kbps) of a track.

The log file entries are followed by a number of queries. Each query is a pair of times (Unix epochs in ms), a, b where $a \leq b$. You should output the total amount of data streamed between times a and b, in kilobits (kb).

Eddie, feeling he is in over his head, has retained you as a consultant to do the actual analysis of the log files.

Input

The first line consists of an integer N, $1 \leq N \leq 100\,000$ which is the number of log entries. The following lines contain integer log entries $t_i\ d_i\ r_i$ as described above.

The log entries are followed by a line with a single integer Q, $1 \leq Q \leq 100\,000$ which is the number of queries. The following Q lines contain time intervals $a_j\ b_j$ as described above.

You may assume that $1\,325\,000\,000\,000 \leq t_i, a_j, b_j \leq 1\,326\,000\,000\,000$, $0 \leq d_i \leq 30\,000\,000$, $1\,325\,000\,000\,000 \leq t_i - d_i$, and $64 \leq r_i \leq 320$

Output

For each of the Q queries a_j b_j, output a line containing the total amount (in kb) streamed during the time interval, rounded to exactly three decimals.

Sample Input 1	Sample Output 1
5 1325338338022 320412 160 1325338361231 441201 320 1325338341231 474123 96 1325338312302 234123 312 1325338331141 623132 147 5 1325300000000 1325400000000 1325338300000 1325338500000 1325338336412 1325338339612 1325338312302 1325338341231 1325338320000 1325338340000	402612.828 38051.567 1588.800 18918.997 12841.247

Problem D
Greedy Cows

Old MacDonald had a farm, and on that farm he had n cows. It is now time to let the cows graze. However, some cows are very greedy and eat the other cows' grass. Being a clever old man, MacDonald decides to split his pasture into several regions. The pasture is surrounded by a great, convex fence.

Photo by Iain McDonald

To split the pasture, he will select a number of points on the surrounding fence, and set up yet more fences between all pairs of selected points, except the points that lie on the same side of the surrounding fence. These new fences may cross. Now, this will create a number of mini-pastures in the pasture (see Figure D.1).

MacDonald wants to minimize the number of points selected (*not* the total fence length, as you might have thought). How many points does he need to select in order to have at least one mini-pasture for each cow? Some of these mini-pastures might be tiny: that doesn't bother Old MacDonald, but only mini-pastures with positive area count.

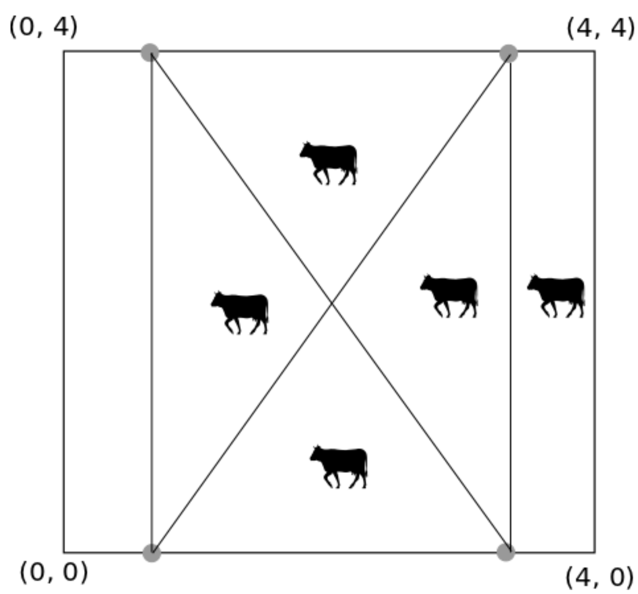

Figure D.1: A possible solution to the first sample case below.

Input

The input begins with an integer $1 \leq n \leq 2^{60}$, the number of cows Old MacDonald has. Then follows an integer $3 \leq F \leq 100$ - the number of fence posts in Old MacDonald's fence. The next F lines will contain two integers $-10^9 \leq C_x, C_y \leq 10^9$, the coordinates of the posts. The F fence posts form a convex polygon with positive area, given in clockwise order.

Output

Output an integer p, the minimum number of points Old MacDonald needs to select, so that each cow gets at least one mini-pasture each.

Sample Input 1	Sample Output 1
5 4 0 0 0 4 4 4 4 0	4

Sample Input 2	Sample Output 2
13 5 1 3 2 3 3 2 2 1 1 1	5

Problem E
Three-State Memory

Photo by thinkpublic

As you might know, the memory in current computers consists of a sequence of bits and each of these bits can be in two possible states. Megan's company has now developed a great new memory unit where every bit has three possible states. This would all be great, if it wasn't for Megan's boss. The boss wants her to write a program for this memory unit for integer operations, but instead of using ternary base (i.e., base 3), Megan's boss decided that the bits should be interpreted as a binary number, with the exception that the digit 2 is allowed as well.

For example, 201 in this strange representation is $2 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9$. Some numbers are shorter in this representation, but for other numbers it doesn't help at all – for instance, 15 is 1111 in binary and this is the only way of writing it in the new representation as well.

It seems that there is nothing else that Megan can do to convince her boss. Since she likes math, she started wondering about different representations of the same number. For example 9 has three representations: $201, 121$ and 1001. Can you help her figure out how many representations a given number has?

Input

The first and only line of the input contains a string consisting of `'0'` and `'1'`. The string represents a non-negative integer N in binary. The leftmost bit is the most significant one. The number of bits will be at least 1 and at most $10\,000$.

Output

Output a line giving the number of different binary representations of N that also use 2 as a digit. Since this number might be big, output the remainder modulo $1\,000\,000\,009$.

Sample Input 1	Sample Output 1
`1001`	`3`

Sample Input 2	Sample Output 2
`1111`	`1`

Sample Input 3	Sample Output 3
00000	1

Problem F
Troll Hunt


Photo by Matthew Hatton

Once upon a time in a land of yore, there was a troll who lived 'neath one of the land's many stone bridges. This troll was quite a mischievous troll, for you see, it had a habit of accusing anyone crossing the bridge of having stolen the troll's property (which was a somewhat curious accusation given that the troll had no property), the punishment of which was to be eaten alive. Unfortunately for the troll, eventually the king got wind of its questionable business model, and sent out the valiant knights of the High Tavern to go, shall we say, Queen of Hearts, on the troll.

Apprehensive of its imminent decapitation, the troll fled, and did not have the decency to even leave a forwarding address. Being a troll, it was clear that the troll was hiding under some other stone bridge than the one it had used for its shady business practice, but which? The knights decided to split up in groups and go search. Since a group needed to be able to avoid being eaten once the troll was found, each group had to consist of at least a certain number of knights. Each group of knights could search under one stone bridge per day (and travelling between bridges was done at lightning speed, thanks to the knights' renowned iTravel$^{\text{TM}}$ technology). While clever enough to flee from its hunting ground, the troll is not bright enough to keep moving between different bridges: once the hunt starts, the troll stays in the same place. How many days would it take until the troll would surely have been found?

Input

The input consists of a single line containing three integers b, k and g, where $2 \leq b \leq 1000$ is the number of stone bridges in the land, $1 \leq k \leq 100$ is the number of knights, and $1 \leq g \leq k$ is the number of knights needed in each group.

Output

Output a line containing a single integer d, the number of days until the troll is sure to have met its destiny.

Sample Input 1	Sample Output 1
`5 2 1`	`2`

Sample Input 2	Sample Output 2
`10 5 2`	`5`

This page is intentionally left blank.

Problem G
Smooth Monkey

Once upon a time there was a Three-Headed Monkey who had a large stash of banana smoothie on one side of the forest. However, her family lived on the other side of the forest. Our dear monkey wants to suprise them and bring them as much banana smoothie as possible.

Photo by wallygrom

Unfortunately, having three heads consumes a lot of energy, so the monkey drinks one millilitre (ml) of smoothie per meter she walks. Note that this is completely continuous, so for example after walking $\frac{\sqrt{2}+\pi}{3}$ meters she has consumed $\frac{\sqrt{2}+\pi}{3}$ ml of smoothie. If she is not carrying any smoothie, she can not move. Furthermore, despite going to the gym every day, the monkey has limited strength and is only able to carry a limited amount of smoothie at a time. Thus she may not be able to bring all the smoothie with her at once.

What she can do however is to leave containers of smoothie anywhere she wants in the forest and then pick them up again later. That way she can for instance move some of the smoothie part of the way, go back to pick up more smoothie, and so on. The monkey starts with all the smoothie in a single place and a (for her purposes) unlimited supply of large containers (large enough so that each of them has room for more smoothie than the monkey can carry). As the monkey only has two hands (despite her large number of heads), she can only carry at most two containers at the same time. The containers are essentially weightless, so the monkey can carry the same amount of smoothie regardless of whether she is carrying one or two containers. They are also bio-degradable, so she can leave empty containers anywhere she likes in the forest without staining her conscience.

How much smoothie (in ml) can she bring to her family on the other side of the forest? The family members are lazy and will not help the monkey transport the smoothie. The monkey does not have to deliver all the smoothie to her family at the same time.

You may make the following somewhat unrealistic (in the sense that no smoothie-transporting monkeys we have ever seen have satisfied them) assumptions:

- The monkey is able to exactly transfer any real number of ml of smoothie between two containers she is carrying. Similarly, she is able to exactly walk any real number of meters.
- Any number of containers can occupy the exact same position (they do so at the start, and during the transport the monkey is able to leave several containers in exactly the same spot).
- Only walking consumes energy: picking up or dropping off a container, turning around, or just standing still, does not consume energy.

Input

The only line of the input contains three integers D, W, C – the distance between the monkey and her family in meters, the total amount of smoothie in ml and finally her maximum carrying capacity in ml. All integers are positive and at most $1\,000\,000$.

Output

Output one line with one number, the amount of smoothie in ml that the Three-Headed Monkey is able to bring to her family. Output will be considered correct if it is within relative or absolute error 10^{-7}.

Sample Input 1	Sample Output 1
`1000 3000 1000`	`533.3333333333`

Sample Input 2	Sample Output 2
`1000 500 1000`	`0`

Problem H
Collapse


Photo by Paul Buckingham

Trouble has come to the remote group of islands known as Insumulia. Due to an unfortunate combination of over-consumption, natural climate variations, and generally difficult conditions, the island of Incunabula has run out of trees. Because several other Insumulian islands depended on trees from Incunabula through trade, its collapse will have repercussions all over Insumulia. In this problem, we'll simulate a (highly oversimplified) model of the situation to determine the effects of the collapse of Incunabula.

We model the situation as follows. Each island has a *threshold* T_i on the amount of incoming goods (for simplicity we assume that there is only a single commodity of goods) it needs to receive per lunar cycle in order for the society of the island to sustain itself. If the amount of incoming goods drops below the threshold, society on the island will collapse and die out, and the island will no longer provide goods to other islands, thereby potentially causing them to collapse as well. Each island provides some amount of goods to a number of other islands. If an island collapses, we assume that goods that would have been delivered to that island is effectively lost; it does not get redistributed and delivered to other islands instead. Also, once an island dies out it is not repopulated (until possibly long after the ongoing collapses have finished).

Your job is to write a program to compute the number of islands that survive after the potential chain reaction of collapses that is caused by the collapse of Incunabula.

Input

The first line of input contains an integer N ($1 \leq N \leq 100\,000$), the number of islands in Insumulia.

Then follow N lines, describing each island. The i'th such description starts with two integers T_i, K_i, where $0 \leq T_i \leq 50\,000$ is the amount of goods the i'th island needs to receive in order to survive, and $0 \leq K_i \leq N-1$ is the number of other islands the i'th islands receives goods from. The remainder of the description of the i'th island is a list of K_i pairs of integers. The j'th such pair, S_{ij}, V_{ij}, indicates that island i receives V_{ij} units of goods from island S_{ij} each lunar cycle. You may assume that the S_{ij}'s are distinct and between 1 and N (inclusive), and that none of them equals i. The values V_{ij} satisfy $1 \leq V_{ij} \leq 1000$ and their sum is at least T_i. The sum of all the K_i's for all the N islands is at most $500\,000$.

Islands are numbered from 1 to N, and Incunabula is island number 1.

Output

Output a single integer, the number of islands surviving the collapses.

Sample Input 1	Sample Output 1
4 0 0 25 3 1 10 3 10 4 10 10 1 2 10 10 1 2 10	0

Sample Input 2	Sample Output 2
4 0 0 20 3 1 10 3 10 4 10 10 1 2 10 10 1 2 10	3