Problem A. Long-Term Mail Storage

Input file: `standard input` or `input.txt`
Output file: `standard output` or `output.txt`
Time limit: 2 seconds
Memory limit: 512 megabytes

Bomboslava is a very busy young lady. She is involved into many different project so the important letters that require an answer overflow her mailbox every day. She typically has no time to answer a letter immediately so she just keeps it unread till she starts to feel guilty. When this happens, she finally starts to answer letters. This happens if at least one of two conditions is satisfied:

- the number of unread letters becomes equal to a;
- the earliest unread letter is already b hours old;

Unfortunately, Bomboslava becomes bored while answering letters so she stops at the middle of the process. If there are exactly k unread letters at the moment she starts to answer letters, she will read and answer exactly $\lceil \frac{k}{2} \rceil$ (half of k, rounded up) oldest of them.

You know that Bomboslava will receive n incoming emails, the i-th of them will drop to the mailbox at the beginning of t_i-th hour, and there will be no two letters coming at the same hour. Each hour Bomboslava starts with checking whether she has any new incoming letters or not, and then she decides whether it's time to start to answer or she can wait a little more.

Input

The first line of the input contains three integers n, a and b ($1 \le n, a \le 100$, $1 \le b \le 1000$) — the number of letters consecutively arriving to Bomboslava's mailbox, the minimum number of unread letters that makes young lady feel guilty, and the minimum number of hours to pass since the earliest yet unread letter arrived that also makes her feel guilty.

The next line contains n distinct integers $t_1, t_2, \ldots, t_n$ in increasing order ($1 \le t_1 < t_2 < \ldots < t_n \le 1000$) — the indices of hours when the corresponding letters drop to Bomboslava's mailbox.

Output

Print n integers, the i-th of them should be equal to the hour during which the i-th letter will be read by Bomboslava.

Examples

standard input	standard output
4 10 100 1 2 3 4	101 101 103 104
8 4 50 1 2 3 4 50 90 91 100	4 4 53 53 100 100 141 150

Note

In the first sample, at the beginning of hour 101 Bomboslava will feel guilty for the first time. At this moment, there will be already 4 letters waiting, i.e. she will read 2 letters. After this, the earliest letter will be received at hour 3 so the next time Bomboslava feels guitly will be at the beginning of hour 103. There will be only 2 unread letter at this moment, so Bomboslava will read only one and stop. Finally, at the beginning of the 104-th hour she will read the last letter waiting.

In the second sample, Bomboslava will feel guilty and answer letters five times.

1. At the beginning of hour 4 the 4-th letter will drop to her mailbox and there will be $a = 4$ letters in total, so she will read first two of them.

2. At the beginning of hour 53 the oldest letter will be $b = 50$ hours old so Bomboslava will read first two of them (3 divided by 2 and rounded up).

3. At the beginning of hour 100 there will be 3 letters waiting with the oldest one coming at the beginning of hour 50. She will check two earliest of them.

4. At the beginning of the 141-st hour there will be two letters waiting: one that came at the beginning of hour 91 and one that came at the beginning of hour 100. The girl will read only one of them.

5. Finally, at the beginning of hour 150 Bomboslava will deal with the last letter.

Problem B. Lassies Versus Machine

Input file:	`standard input` or `input.txt`
Output file:	`standard output` or `output.txt`
Time limit:	2 seconds
Memory limit:	512 megabytes

Dusya and Lusia came to the Competitive Programming Workshop in Retropolis. The dorm they live in has a vending machine that sells coffee and different snacks. The problem is that it accepts only coins of 1, 5 and 10 burles, and banknotes of 50 and 100 burles. The banknotes of 500, 1000 and 5000 burles are not accepted by this vending machine. There are no other banknotes or coins in Berland.

Dusya and Lusia have enough money but they use the machine every day so they are always out of the proper coins and banknotes. Today they spent a lovely evening in the place called "Soda&Sandwiches" with a total check of n burles. Girls have decided to split the bill in two parts, each of at least a burles. Their goal is to receive as a change the set of banknotes and coins with the maximum possible total value of banknotes and coins that can be used in the vending machine. Check of size x is paid according to the following algorithm:

1. The girl who pays this check gives a waiter minimum number of 5000 banknotes that make the sum of at least x. Let this number be equal to k.
2. Waiter gives her back the change of size $5000k - x$ using the minimum possible number of banknotes and coins. One can easily prove that such a set is uniquely determined for any value of a change.

Please, help Lusia and Dusya split the check of n burles in two checks of integer values x and $n - x$, each greater than or equal to a, such that the total value of the banknotes and coins suitable for using in the vending machine in the **change** is maximum possible.

Input

The only line of the input contains two integers n and a ($1 \le 2 \cdot a \le n \le 10^{18}$) — amount of money to be paid and the minimum amount each of the girls should contribute.

Output

Print one integer — the maximum "useful" amount of change, i.e. the total sum of banknotes and coins that can be used in the vending machine.

Examples

standard input	standard output
1000 500	0
10000 4000	500
623 57	877

Note

In the first sample, the only possible way to split the check is for each of the girls to pay 500 burles. Each of them will receive a change of $5000 - 500 = 1000 + 1000 + 1000 + 1000 + 500$ burles. All these banknotes can not be used in the vending machine.

In the second sample, girls can split 10 000 burles in parts of size 4200 and 5800 burles respectively. The first change will be equal to $5000 - 4200 = 500 + 100 + 100 + 100$ burles, while the second change will be $10\,000 - 5800 = 1000 + 1000 + 1000 + 1000 + 100 + 100$, thus Dusya and Lusia will receive five banknotes of 100 burles that can be spent in the vending machine.

Problem C. Efficient Management Returns

Input file:	`standard input` or `input.txt`
Output file:	`standard output` or `output.txt`
Time limit:	3 seconds
Memory limit:	512 megabytes

While human resources managers do their best to make the engineers feel comfortable and work better they sometimes have to solve pretty weird problems. One well-known IT-company "Soft Ware Engine Near" has n software engineers numbered from 1 to n. HR knows about m pairs of engineers that dislike each other. Managers would like to pick some value k and distribute all engineers between k teams and obey the following rules set by the CEO:

1. Everyone should feel comfortable at his workplace, so there should be no two engineers who dislike each other in one team.
2. The distribution should be efficient, that means there should be no two teams that can be united in one with obeying rule 1.

Note that you can pick any value of parameter k, you are not asked to maximize or minimize it.

Input

The first line of the input contains two integers n and m ($1 \le n \le 200\,000$, $0 \le m \le 200\,000$) — the number of engineers in the company and the number of pairs of engineers who dislike each other.

Each of the following m lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) — the indices of the engineers in the corresponding pair. It's guaranteed that no pair occurs twice.

Output

In the first line print the number of teams k. The second line should contain n integers $c_1, c_2, \ldots, c_n$ from 1 to k inclusive, that mean the i-th engineer should be assigned to the team c_i. Each team from 1 to k should contain at least one engineer, no two engineers that dislike each other should be assigned to the same team, and it should not be possible to unite two team in one without breaking the previous rule.

Examples

standard input	standard output
3 2 1 2 2 3	2 1 2 1
5 6 1 2 2 3 1 3 5 4 5 3 4 3	3 1 2 3 1 2

Problem D. The Sting

Input file:	`standard input` or `input.txt`
Output file:	`standard output` or `output.txt`
Time limit:	2 seconds
Memory limit:	512 megabytes

Recently Ostap Bender (famous Russian fictional character) came up with a 401-st relatively legitimate way of acquiring money from people, and, what is most important: it is absolutely safe! He noticed that people of Stargorod are huge fans of a local team "Star Dogs" in a Professional Football League of Fictional Cities, and they love making bets on football game outcome.

Ostap has already received n proposals of bets on outcome of the next game. Each of the proposers chooses one of the three possible game outcomes: either favourite Stargorod team wins, loses or there will be a draw. In case game outcome matches the chosen, Ostap has to pay the bet proposer a_i burles. In case of any other game outcome the bet proposer pays Ostap b_i burles.

Ostap's plan is to accept some of the proposals he received and decline the remaining proposals. Ostap wants to choose such subset of proposals that maximizes his profit **in the worst case**.

Input

The first line of the input contains one integer n ($1 \le n \le 500$), the number of bet proposals Ostap received.

Each of the following n lines contains one bet description, consisting of a character t_i and two integers a_i and b_i ($1 \le a_i, b_i \le 500$), where t_i is "`W`", "`D`" or "`L`" depending on if the proposer believes in win, lose or draw of a Stargorod favorite tea, a_i is amount of money Ostap has to pay in case chosen outcome happens and b_i is amount of money the proposer has to pay Ostap in remaining cases.

Output

Output the maximum possible worst-case profit Ostap can have if he chooses the optimal subset.

Examples

standard input	standard output
3 W 2 2 D 2 2 L 2 2	2
3 W 3 1 D 3 1 L 3 1	0
5 W 3 3 W 2 2 D 10 100 L 3 1 D 3 3	1

Problem E. Random Value of Mode

Input file:	`standard input` or `input.txt`
Output file:	`standard output` or `output.txt`
Time limit:	4 seconds
Memory limit:	512 megabytes

Despite the fact it is still snow in Moscow, and it's not clear whether the summer is going to happen at all, Gleb wants to buy fancy summer clothes and that's why he decided to buy a trench coat. He got into a huge mall consisting of n shops arranged in a long line. Shops are numbered from 1 to n in order the are located along the line. For all i between 1 and $n-1$ shops i and $i+1$ are adjacent, and moving between them takes exactly one second. Gleb has just parked his car near the shop number x and he is going to look for the desired trench coat in this mall.

For each shop Gleb knows that in case he visits this shop, he will spend exactly t_i seconds to find out if there is a desired coat there or not. It is known that the shop i will have a coat with probability of p_i percents. Coat may be found in each shop independently from any other shop, in particular, desired coat may be in several shops as well as in none of them. Gleb visits each shop at most once, and if Gleb passes by the shop he hasn't yet visited, he **always visits** it and spends time needed to find a coat in it, even if he knows for sure there is no chance to find the coat there, i.e. $p_i = 0$.

Since Gleb is not a fan of visiting shopping malls, he immediately buys the coat as soon as he finds it and returns back to the shop number x. Now he is interested, if he use the best possible strategy at any moment, what is the minimum possible expected time he spends in the mall? Note that Gleb leaves the mall only when he finds the desired coat or when there are no unvisited shops left.

Input

The first line of the input contains two integers n and x ($1 \le x \le n \le 300\,000$), the number of shops in a mall and the shop near which Gleb has parked.

In the i-th of the following n lines there are two integers p_i and t_i ($0 \le p_i \le 100$, $1 \le t_i \le 1000$), the probability of finding the desired coat in the i-th shop, given as an integer number of percent, and the number of seconds Gleb has to spend while looking for a coat in this shop.

Output

Print one real number, equal to the expected number of seconds Gleb will leave the mall if he uses the optimum strategy of visiting the shops. Your answer will be considered correct if its absolute or relative error doesn't exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1,b)} \le 10^{-6}$.

Examples

standard input	standard output
5 1 50 1 50 1 50 1 50 1 50 1	3.8125000000
4 2 1 1 0 10 0 10 100 1000	1016.8600000000

Problem F. Measure Twice, Divide Once

Input file:	`standard input` or `input.txt`
Output file:	`standard output` or `output.txt`
Time limit:	2 seconds
Memory limit:	512 megabytes

Bomboslav's favorite principle when solving programming contest problems is "Divide and Conquer", though he believes that for some trees this principle may be used in a more efficient way than provided by centroid decomposition (*you do not need to know this term in order to solve this problem*).

Bomboslav wants to be able to determine the minimum possible recursion depth when using the "Divide and Conquer" strategy with trees. Let's describe the procedure formally. We are given a tree consisting of n vertices. While we have at least one vertex in our graph, repeat the following procedure:

1. in each connected component choose exactly one vertex;
2. remove chosen vertices and all edges that are adjacent to them.

What is the minimum number of steps needed for removing all the vertices of the given initial tree when using aforementioned procedure?

Input

First line of the input contains an integer n ($1 \le n \le 300\,000$), the number of vertices in Bomboslav's tree.

The i-th of the following $n-1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$), vertices that are connected by the i-th edge.

Output

Output the only integer, minimum number of steps of the given procedure that is needed in order to delete all the vertices of the given tree.

Examples

standard input	standard output
3 2 3 1 3	2
11 1 2 2 3 3 4 3 5 3 6 4 7 4 8 4 9 4 10 4 11	3