ACM ICPC 2010-2011 NEERC Moscow Subregional Contest
Moscow, October 24, 2010

List of Problems

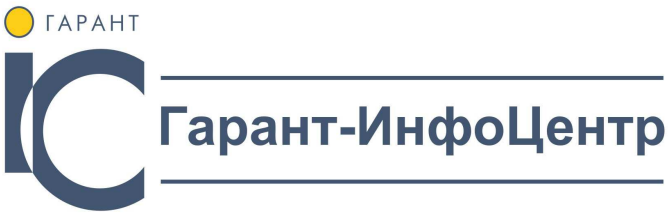
Problem	Page	Time limit	Memory limit	Name
A	2	1 second	256 megabytes	Alien Visit
B	3	3 seconds	256 megabytes	Big Number
C	4	1 second	256 megabytes	Contest
D	5	1 second	256 megabytes	Distance
E	6	1 second	256 megabytes	Efficient Cartography
F	7	1 second	256 megabytes	Finance
G	8	1 second	256 megabytes	Golden Spire
H	10	1 second	256 megabytes	Hometask
I	11	2 seconds	256 megabytes	Interest Targeting
J	13	1 second	256 megabytes	Joke
K	14	2 seconds	256 megabytes	KMC Attacks
L	15	2 seconds	256 megabytes	Lanes

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.

Sponsors



Problem A. Alien Visit

Time limit: 1 second
Memory limit: 256 megabytes

25 May, 1997, near Elcino-Borisovo place, Jandal region, strange signs were found in the field...

Witness: "First, I saw only one UFO. It was shining with cold-blue light. Closer to the center of the object, the light was pink. It hung over the field, then began to blink and move intermittently round. The UFO was quite big. The second UFO came several minutes after the first. It had the same size as the first one. It seemed that there was some kind of contact between them — they began to blink alternately."

Circles of scorched barley were found in the field. The circles were of the same radius, and their centers were lying on a straight line.

You were hired to investigate the damage caused to the farms of Elcino-Borisovo place by the visit of aliens. In order to do this you are to calculate the total area of scorched barley.

Input

The first line of the input contains two integers n and r denoting number of circles and the radius of the circles, respectively ($1 \le n \le 1\,000$, $1 \le r \le 100$). The next line contains n space separated integers $a_1, a_2, \ldots, a_n$ — the shifts of circles' centers relative to some origin ($0 \le a_i \le 5\,000$). All shifts are guaranteed to be distinct.

Output

Output the only real number — the total area covered by these circles. The relative error of your answer must not exceed 10^{-6}.

Examples

stdin	stdout
1 1 0	3.1415926536
2 2 0 2	20.2192624343

Problem B. Big Number

Time limit: 3 seconds
Memory limit: 256 megabytes

Trillion, quadrillion, quintillion, sextillion, septillion, octillion — all is trifle! Vasya knows how to get a really big number. One should consider several "ordinary" numbers, and then write them down consequently one after another on a piece of paper.

Vasya decided to demonstrate this idea to his elder brother Pete. He prepared a sequence of N positive integers and started writing them down one after another on a piece of paper.

After observing this process for a while, Pete got tired and suggested Vasya to solve the following puzzle. Vasya's goal is to remove exactly K numbers from his sequence so that if one concatenates the remaining numbers (keeping the original order) he gets the largest possible number.

Please, help Vasya.

Input

The first line contains two integers N and K denoting how many numbers are prepared by Vasya and how many of them should be removed, respectively ($2 \le N \le 10^5$, $1 \le K \le N - 1$ and $K \le 100$). The second line contains N space separated positive integers not exceeding 10^9.

Output

Please, output the largest possible number Vasya can obtain by removing exactly K numbers from the sequence.

Examples

stdin	stdout
5 2 123 37 45 9 18	1234518

Problem C. Contest

Time limit: 1 second
Memory limit: 256 megabytes

The second round of the annual student collegiate programming contest is being held in city N. To be prepared for the inrush of participants, the jury needs to know the number of them attending the previous, first round.

Unfortunately, the statistics regarding that first round (including the final standings) was lost during a recent disk failure and no backup was made.

The only hope is a short statistical summary that was found written on a tiny piece of paper by the oldest jury member. The percentage of teams which have solved the problem is provided for each problem of the the first round. Each percentage is an integer rounded using the usual mathematical rules (numbers with a fractional part strictly less than .5 are rounded down, the others are rounded up).

This is the only information the jury has at hand. Also, that oldest jury member clearly remembers that a prize was awarded to some team during the first round, probably for winning it. Hence, at least one team had participated in the first round.

Input

The first line of input contains an integer N ($3 \le N \le 12$) — the total number of problems in the contest. The second line of input contains N integers $P_1, \ldots, P_N$. Each number P_i ($0 \le P_i \le 100$) denotes a percentage of the teams solved the i^{th} problem.

Output

Print out the minimum possible number of teams that could have participated in the first round.

Examples

stdin	stdout
3 33 67 100	3

Problem D. Distance

Time limit: 1 second
Memory limit: 256 megabytes

In a large city a cellular network operator is holding a competition for subscribers to promote their new "pedestrian navigator" service. The main prize will be awarded to the first pair of subscribers to meet each other. The competition ends when any such meeting takes place.

At the start of the competition all the subscribers are at their known positions, are able to see each other on their smartphones, and are moving at a constant speed of 10 km/h taking only pedestrian walks. Each subscriber is willing to win the prize and is indifferent to the others.

In order to prepare for an award ceremony the cellular network operator needs to know the minimal amount of time after which the competition may come to an end.

Input

In the first line of input integers N, K, and L are given — the number of subscribers in a cellular network company ($2 \le N \le 10^5$), the number of junctions ($1 \le K \le 10^5$), and the number of pedestrian walks ($1 \le L \le 10^5$) in the city, respectively.

On the next N lines of input S_i ($1 \le S_i \le K$) numbers are given — initial positions of subscribers (in the terms of transport graph junctions).

The next L lines of input pedestrian paths are given in the form of integers B_i, C_i and D_i separated by spaces. Each line denotes that there is a two-way pedestrian path between junctions B_i and C_i ($1 \le B_i, C_i \le K$, $B_i \ne C_i$) with a length of D_i ($1 \le D_i \le 5000$) kilometers.

Output

Output the minimal possible number of minutes that may elapse from the start till the end of the contest. It is guaranteed that at least one pair of the subscribers can meet.

Examples

stdin	stdout
2 2 1 1 2 1 2 5	15
3 3 3 1 2 3 1 2 4 3 2 4 3 1 4	12
2 3 3 1 2 1 2 9 3 2 5 1 3 3	24

Problem E. Efficient Cartography

Time limit: 1 second
Memory limit: 256 megabytes

During the digitization of the Yandal region map it has become known that it represents a connected area in its bitmap pixel form and is wholly contained within a $M \times N$ pixel rectangle. Also, it has the following unique feature: each horizontal or vertical line intersects the region by exactly one segment.

The map was approved for use in portable navigation devices. The screen size of a typical navigator device is $A \times B$ pixels. In order to display this particular map the plane should be divided into rectangles of $A \times B$ pixels each by horizontal and vertical lines passing by pixel boundaries. Then, each such rectangle (called a *tile*) containing at least one pixel of map's area is loaded into the device.

Your task is to choose the positioning of these separating lines for a given map to minimize the total number of tiles to be loaded into the device. You are not allowed to rotate the map or the screen.

Input

The first line contains 4 integers — M (the number of lines), N, A and B.

Each of the next M lines contains two numbers L_i and R_i each — the leftmost and the rightmost index of the column containing region's pixels on the i^{th} horizontal bitmap pixel line.

The limits for possible values are: $1 \le M, N \le 10^5$, $1 \le A, B \le 1\,000$, $1 \le L_i \le R_i \le N$.

Output

Output the only integer — the minimal possible number of map tiles.

Examples

stdin	stdout
5 7 2 2 1 2 1 3 2 4 3 5 4 5	5
7 10 3 4 4 7 4 7 1 10 1 10 1 10 4 7 4 7	5

Problem F. Finance

Time limit: 1 second
Memory limit: 256 megabytes

The Big Boss Company (BBC) principal is going to prepare a vacation schedule for the next year. Each employee of BBC gets a four-week vacation each year. A vacation can be divided into several parts. Each part consists of several weeks. The parts are separated by at least one work week.

According to the law, the pay for each vacation part is calculated separately from the others as follows. Consider the employee's salary for the period of 52 weeks preceding the current part of the vacation and divide it by 52. (A year is assumed to consist of exactly 52 weeks.) All vacation pays during these 52 weeks are ignored for the purpose of salary calculation. The result is multiplied by the number of weeks in this part of the vacation (ranging from 1 to 4).

Given the vacation schedules for the current year, you should help the BBC principal to construct the next year's schedule that minimizes the total vacation pay costs. Since the pays for distinct employees are unrelated you are to solve the problem assuming that there is only one employee.

Input

The input contains four integers listed in the ascending order giving the vacation weeks in the current year's schedule. Weeks are specified by numbers ranging from 1 to 52.

Output

Output the numbers of vacation weeks for the next year in the ascending order. The vacation schedule should minimize the vacation pay costs. If there are several equivalent solutions, output any of them.

Examples

stdin	stdout
2 3 20 21	1 2 3 5

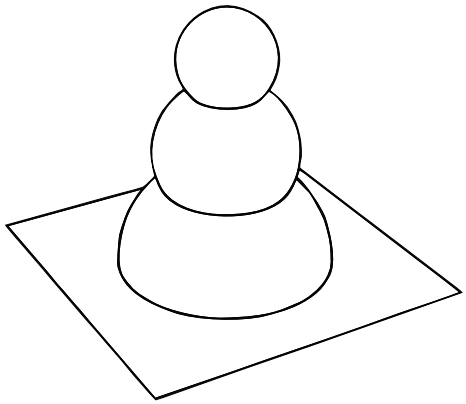
Note

The calculation of the pay for the first part considers $52 - 4 = 48$ work weeks while the calculation of the second part considers $52 - 5 = 47$ work weeks.

Problem G. Golden Spire

Time limit: 1 second
Memory limit: 256 megabytes

There are some peculiar facts known about bacterium "Jandelinka Fest", which inhabits the Jandal region. When a strain of this bacteria is exposed to air, it starts propagating uniformly in all directions so that the boundary of its population is an ideal sphere centered in the initial place of the strain habitation. The radius of the sphere is linearly increasing over time.



Moreover, bacteria do not tolerate the scent of certain aroma, so if being smoked they immediately die. This property is often used by the local ministers of religion. They decorate the temples with golden spires of fanciful forms as follows. An ideally straight thin vertical stick is set up on a flat horizontal ground surface. Some points of the stick are covered by bacteria strains (different strains are placed in different points). As different colonies may be of different bacteria species, they may have different rates of propagation. After a week monks stop the process by smoking the stick, and cover the resulting construction by golden dust. Then the spire is set up on a peak of a temple.

The spheres corresponding to distinct colonies may collide but the species are completely indifferent to each other. Hence, a point is covered by the colonies if and only if it is contained in any of the above spheres (all growing independently). Also, when bacteria reach the ground level (corresponding to zero height) they die immediately.

You are asked to find total area of the spire surface treating the stick as an ideal line. The reason is monks should estimate the amount of gold needed to finish the spire. Only spherical parts of the surface should be counted.

Input

The first line contains an integer M — the number of colonies ($1 \le M \le 2\,000$). The next M lines contain two single space separated real numbers each. The first (H_i) defines the height of the colony over the surface (given in feet). The second (V_i) defines the rate of propagation (the speed at which the radius increases, in inches per hour). $0 \le H_i \le 1000$, $0 \le V_i \le 10$.

Output

The only line of the output should contain the total area of the spherical parts of the surface, in square feet. The relative error of your answer must not exceed 10^{-6}.

Examples

stdin	stdout
1 0 1	1231.504320
2 0 1 7 1	1847.256480
2 10 2 24 1	6685.309167

Note

1 foot = 12 inches, 1 week = 7 days.

Problem H. Hometask

Time limit: 1 second
Memory limit: 256 megabytes

Kolya is still trying to pass a test on Numbers Theory. The lecturer is so desperate about Kolya's knowledge that she gives him the same task every time.

The problem is to check if $N!$ is divisible by N^2.

Input

The first line of input contains the only integer N ($1 \le N \le 10^9$).

Output

Please, print to output "`YES`" provided that $N!$ is divisible by N^2, otherwise print "`NO`".

Examples

stdin	stdout
3	NO
9	YES

Note

$N! = 1 \cdot 2 \cdot \ldots \cdot N$

Problem I. Interest Targeting

Time limit: 2 seconds
Memory limit: 256 megabytes

A unique display advertisement system was developed at the department of advertising technologies, Yaagl Inc. The system displays advertisements that meet the interests of the user who is currently watching the page.

For this system to function properly, having a good method for computing the user's category of interest and the probability of clicking an advertisement, which is related to his/her interests, is vital.

One of your colleagues has implemented an algorithm that analyzes users' browsing history and produces the results as follows:

user_id	category_id	create_time	heuristic_ctr

where:

- `user_id` is the user identifier;
- `category_id` is the identifier of user's predicted interest category;
- `create_time` is the prediction generation time;
- `heuristic_ctr` is the predicted click probability for this category.

This information is stored in `interests_log` table.

Your task is to write a program which estimates the prediction quality. You are provided with log table `events_log` containing advertisement display results. Each row of the table corresponds to advertisement display event. The table has the following columns:

- `user_id` is the user identifier;
- `category_id` is the identifier of an advertisement category;
- `adv_id` is the advertisement identifier;
- `show_time` is the advertisement display time;
- `click_flag` is 1, if a click had occurred, 0 otherwise.

Your are expected to add new information from the first table to the second one, or, as SQL-developers usually say, do an `INNER JOIN` of these two tables using (`user_id`, `category_id`) as a key.

While performing the join, the following conditions must be satisfied:

- `user_id` and `category_id` of matching rows must be equal;
- each row of the second table can match at most one row of the first table;
- for a pair of matching rows the following must hold — `show_time` $>$ `create_time` and `show_time` $-$ `create_time` is minimum.

All matching rows must appear in the result. However some rows from both tables may not appear in the result if they have no match.

Input

The first line contains the numbers `interests_count` and `events_count`, denoting the sizes of the log tables `interests_log` and `events_log` respectively. The sizes do not exceed 70 000. The next `interests_count` lines contain rows of `interests_log`, and the next `events_count` lines contain rows of the second table. Field values are separated by a space. All field values except for `click_flag` are integers belonging to the range $[1, 10^9]$. For the records in `interests_log`, all the tuples (`user_id`, `category_id`, `create_time`) are unique.

Output

Output the joined table. Each row should be as follows:

user_id	category_id	create_time	heuristic_ctr	adv_id	show_time	click_flag

Print the number of rows in the first line. Then print table rows, one per line. Order the rows by tuples (`heuristic_ctr`, `user_id`, `category_id`, `create_time`, `adv_id`, `show_time`) in the ascending order. Tuples are compared lexicographically, i.e. tuples are compared first by `heuristic_ctr`, then by `user_id` and so on till `show_time`. You can output rows in any order satisfying the described criteria.

Examples

stdin	stdout
2 2 1 1 102 200 2 1 104 333 2 1 33 101 0 1 1 34 105 1	1 1 1 102 200 34 105 1
6 7 3 88 210 1000000 1 99 210 2000000 2 88 110 3000000 2 88 210 4000000 3 88 310 5000000 3 75 100 6000000 3 88 1001 310 0 3 88 1002 211 1 1 99 1003 210 0 2 88 1004 100 0 2 88 1005 210 1 2 88 1006 310 0 3 75 1007 331 1	5 3 88 210 1000000 1001 310 0 3 88 210 1000000 1002 211 1 2 88 110 3000000 1005 210 1 2 88 210 4000000 1006 310 0 3 75 100 6000000 1007 331 1

Problem J. Joke

Time limit: 1 second
Memory limit: 256 megabytes

The problem is to cut the largest possible number of circles with diameter y out of a stripe of length x and width y.

Input

The only line of input consists of two positive real numbers x and y with 9-digit precision separated by spaces. The integers may be written without decimal point.

Output

Output a single integer — the maximum number of circles one can cut out of the stripe.

Examples

stdin	stdout
6.3 0.9	7
0.63 0.09	7
1 1	1

Problem K. KMC Attacks

Time limit: 2 seconds
Memory limit: 256 megabytes

Warrant VI is a remote planet located in the Koprulu Sector. Warrant VI features a strange huge field, which was discovered by the Kel-Morian Combine (KMC). The KMC is involved into a conflict with the Scientific Community, which wishes to study the field, but the KMC intends to hold on to the significant mineral wealth located there.

The field is represented by a rectangle of size $N \times M$ cells. All coordinates are positive integers. The left upper corner is located at $(1, 1)$ and the right bottom corner is at (N, M). The Scientific Community have sent a T-280 unit to the field. You are one of the KMC members, your task is to find the unit and destroy it. You have an orbital base equipped with a scanner at your disposal. You can examine any single cell and if scan reveals the unit, a detachment is sent to destroy it. After every scan the unit changes its location (i, j) by moving into one of four (three if the unit is in a border cell or two if it is in a corner cell) adjacent cells. The unit has a tracking device. For every unit's move you receive a signal: '`U`' (for up, i decreases by one), '`D`' (for down, i increases by one), '`L`' (for left, j decreases by one), or '`R`' (for right, j increases by one).

Time is running out so you are allowed to make not greater than 50000 scans to reveal the T-280 location. Although the initial position of the unit is unknown, each its move provides you with information. You are forbidden to scan cells, which definitely cannot hold the unit according to the gathered information.

Input

This is an interactive problem. Your program should first read a line with two integers N and M from the standard input (where N is the height and M is the width of the field, $1 \le N, M \le 200$). Then, for every scan your program should print a corresponding request to the standard output (see below) and read a reply from the standard input. Each reply is placed in a separate line and is either '`U`', '`D`', '`L`', '`R`' (meaning that you have missed and the unit has moved in the given direction) or '`DETECTED`' (meaning that your last scan has revealed the unit's position).

Output

For each scan print its coordinates i, j in a separate line (where i is the row number and j is the column number). You have to flush the standard output after printing each request.

Examples

stdin	stdout
1 2 L DETECTED	1 1 1 1

Note

The output pipe from your program to the interactor program and the input pipe back have limited capacities. Your program must follow the above-described protocol to avoid deadlock. Deadlock condition is reported as a time-limit exceeded error.

To flush the standard output stream use the following statements:

In C use `fflush(stdout);`	In C++ use `cout.flush();`	In Java use `System.out.flush();`

Once the unit is detected, your program must close the standard output stream and terminate with zero exit code.

Problem L. Lanes

Time limit: 2 seconds
Memory limit: 256 megabytes

There are S people swimming in the pool.

At some point, the coach notices that their distribution over pool lanes is quite uneven, which is a nuisance for pool visitors. Changing the lane across ropes, however, poses a nuisance for a swimmer.

To form a nice distribution over the lanes the number of swimmers on each pair of adjacent lanes should differ by at most one. In particular, an empty lane should only be adjacent to another empty one or a lane occupied by only one swimmer.

Your task is to calculate the minimum number of swimmers' lane changes needed to meet this condition.

Input

The first line of the input contains the only integer N — the number of lanes, $1 \le N \le 400$. The next line contains N numbers separated by spaces: the number of swimmers in the first, second, ..., N^{th} lane respectively. The sum of these numbers equals S ($0 \le S \le 1\,500$).

Output

Output the minimum possible number of lane changes that are needed to meet a desired distribution.

Examples

stdin	stdout
3 8 0 2	5
3 8 5 7	1