

ТУЙМААДА XXX Международная олимпиада
ТУУМААДА The 30th International Olympiad



PROBLEMS AND SOLUTIONS
INFORMATICS
ИНФОРМАТИКА
ЗАДАЧИ И РЕШЕНИЯ

Yakutsk 2023
Якутск 2023

XXX Международная олимпиада школьников по математике, физике, химии и информатике «Туймаада». ИНФОРМАТИКА. – Якутск, 2023.

Сборник содержит задачи XXX Международной олимпиады «Туймаада» по информатике, а также возможные варианты решений. Олимпиада проводилась 2–10 июля 2023 года в г. Якутске. Олимпиада прошла в смешанном очном и дистанционном форматах в два соревновательных дня, в каждый из которых участникам было предложено решить по пять задач.

This booklet contains the problems and possible solutions of the 30th Tuymaada International Olympiad in informatics. The Olympiad was held on July 2–10, 2023 in Yakutsk, Russia. The Olympiad was held both remotely and on-site, in two competition days. The participants were invited to solve five problems on each day.

АВТОРЫ AUTHORS

Виталий Витальевич ИВАНОВ Ай-Новус	A	<i>Vitalii IVANOV I-Novus</i>
Сергей Геннадьевич ВОЛЧЁНКОВ Ярославский государственный университет им. П. Г. Демидова Соавторы: Василий ПАРНИКОВ Дьулустан НИКИФОРОВ	B	<i>Sergei VOLCHENKOV P. G. Demidov Yaroslavl State University Co-authors: Vasilii PARNIKOV Dulustan NIKIFOROV</i>
Денис Иванович МАКАРОВ студент, ИТМО	C	<i>Denis MAKAROV student, ITMO</i>
Дьулустан Васильевич НИКИФОРОВ СВФУ	D	<i>Dulustan NIKIFOROV NEFU</i>
Василий Васильевич ПАРНИКОВ студент, ИТМО	E	<i>Vasilii PARNIKOV student, ITMO</i>
Сергей Геннадьевич ВОЛЧЁНКОВ Ярославский государственный университет им. П. Г. Демидова Соавтор – Дьулустан НИКИФОРОВ	F	<i>Sergei VOLCHENKOV P. G. Demidov Yaroslavl State University Co-authored by Dulustan NIKIFOROV</i>
Василий Алквадovich БУЛАТОВ Яндекс Облако	G	<i>Vasilii BULATOV Yandex Cloud</i>
Тимур Алексеевич ЗЫКОВ студент, МФТИ	H	<i>Timur ZYKOV student, MIPT</i>
Никифор Никитич ПАВЛОВ СВФУ	I	<i>Nikifor PAVLOV NEFU</i>
Тимур Алексеевич ЗЫКОВ студент, МФТИ	J	<i>Timur ZYKOV student, MIPT</i>

Problem statements

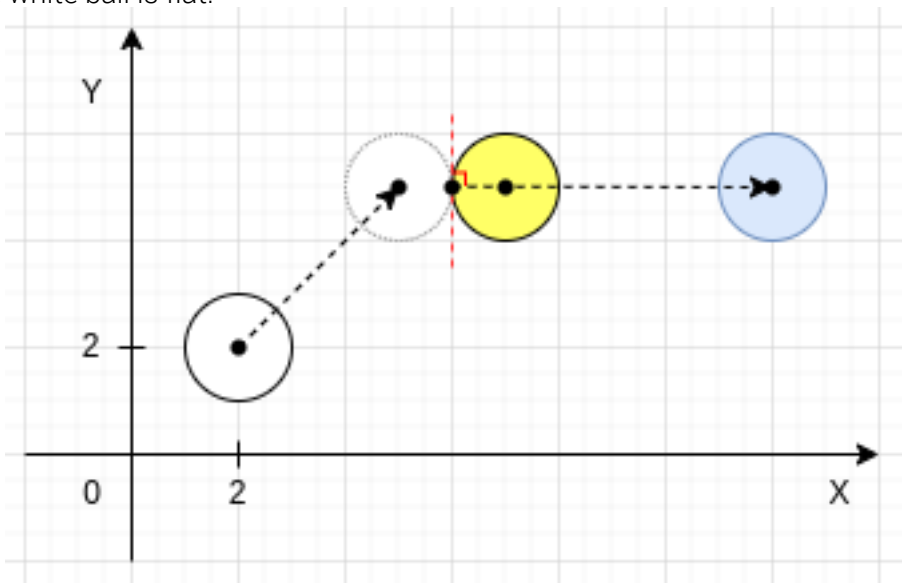
A. Golfilliard

Time limit: 1 second

Memory limit: 256 megabytes

The programmer Petya, after a devastating loss in golfilliard, decided to create a program that could calculate how to hit the white ball so that the center of colored one hit exactly in the center of the pocket.

Of course, in his virtual golfilliard game, friction and air resistance are completely absent, the collision of the balls is absolutely elastic, and the impact on the white ball is flat.



Input

The first line contains the coordinates of the center (X_1, Y_1) of the white ball. In the second line – the coordinates of the center (X_2, Y_2) of the colored ball. In the third line – coordinates (X_3, Y_3) of the center of the pocket. All numbers are integers and range from -1000 to 1000.

Radiuses of the balls are the same and equal to 1.

Output

Output the angle α in degrees relative to the X-axis counterclockwise, in the direction of which the white ball should be hit ($0 \leq \alpha < 360$). Output the answer with an accuracy up to fourth order.

It is guaranteed that the answer exists.

Example

standard input	standard output
2 2 7 5 12 5	45.0000

Note

In a flat shot, the ball being hit slides without any rotation before it hits the object ball.

B. The librarian-robot

Time limit: 1 second
Memory limit: 256 megabytes

The library has hired an artificial intelligence-equipped robot to move books from shelf to shelf. It can perform only one action – remove any number of consecutive books from shelf A and place them to another one (besides the shelf A , there are K shelves numerated from 1 to K). All remaining books on shelf A are then compressed, i.e. moved to one side.

The robot did a good job the day before and the shelf A is empty now. However, today another batch of books was brought to the library and placed on shelf A . Each book is marked with a number, indicating which shelf it should be moved to. You are asked to develop a program that will compute the least number of actions to move all the books into their corresponding shelves.

Input

The first line contains integer $1 \leq N \leq 200$ – a number of new books.

The second line contains integer $1 \leq K \leq 200$ – a number of different shelves that are numbered from 1 to K .

The third line contains N space-separated integers a_1, a_2, \dots, a_N – shelf number of the given books.

Output

Your program should output one integer – the minimal number of robot's actions to place books to corresponding shelves.

Note

The scores for each subtask are awarded if and only if all tests for that subtask and its required subtasks are successfully passed.

Subtask	Add. constraints	Score	Req. subtasks
1	$K \leq 1$	7	–
2	$K \leq 2, N \leq 20$	8	1
3	$K \leq 2$	14	1, 2
4	$N, K \leq 10$	6	–
5	$K \leq 10$	25	1, 2, 3, 4
6	None	40	1, 2, 3, 4, 5

Example

standard input	standard output
6 3 2 1 1 3 1 2	3

Note

In the example the robot first puts the 4th book on shelf 3. Then books with indexes 2, 3, 4 on shelf 1. Then books with indexes 1, 2 on shelf 2.

C. I'm the Map

Time limit: 1 second
 Memory limit: 256 megabytes

Dora the Explorer borrowed books from the library. And now she had finished reading and need to return them before the library closes. But here's the

Problem statements

problem — Dora doesn't know the path to the library from the place where she is now.

Dora needs your help. There is the Map in her backpack. Say «Map!» and find a path to the library.

The Map is an undirected tree, the root of this tree is place where Dora is. Each vertex is numbered from 0 to $n - 1$. Consider two functions:

1. $f(x)$, which takes a vertex number and returns a vertex number equal to $x \oplus p[x]$ modulo the tree size, where $p[x]$ is the parent x . The parent of the root is the root itself.
2. $g(x)$ takes a vertex number and returns a vertex number equal to $x \mid p[x]$ modulo the tree size.

The address (number of vertex) of the library can be calculated as follows $f(g(f(g...(x)...))$. These functions alternate and are executed k times in total. The functions' chain starts with the function f (which means that the most outer function is the function f).

Please, help Dora to find the shortest path to the library.

Input

The first line contains three numbers n, k, s — the number of tree vertices, the number of function iterations, and the number of vertex where Dora is. ($0 < n \leq 10^5, 0 < k \leq 10^{18}, 0 \leq s < n$)

Next $n - 1$ lines contain edges through pairs of vertex numbers a and b ($0 \leq a, b < n$)

The last line contains one number x - the number of the starting vertex of functions' chain.

Output

On the first line print «Map!» without quotes.

On the second line print the path (space separated numbers of vertices) from the root to the library.

Note

All tests are scored independently.

Examples

standard input	standard output
<pre>7 2 2 0 1 5 1 1 6 2 0 2 4 2 3 6</pre>	<pre>Map! 2</pre>
<pre>7 3 6 4 3 2 1 0 1 3 1 2 6 2 5 4</pre>	<pre>Map! 6 2 1 3</pre>

Note

1. $a \oplus b$ is a bitwise XOR of two numbers.
2. $a | b$ is a bitwise OR of two numbers.
3. Tree is an acyclic connected graph.
4. Tree size is the number of vertices in the tree.

D. A game for the gamer-student

Time limit: 1 second
 Memory limit: 256 megabytes

Many IT students like playing computer games.

The teacher of programming suggested the following game: there is a $2 \times N$ field (two rows of N cells) populated with monsters in some cells. You have only one ability: in one move to burn three consecutive cells to the

Problem statements

ground; the monsters in those cells die. So in one turn you can either burn a vertical row of two cells, or a horizontal — of three (or less if you burn at the edge of the field).

Of course, the teacher expects you to write a program that will completely clear the field of monsters in the *least* number of moves.

Input

The first line contains a single number N ($0 < N \leq 3 \cdot 10^5$).

The second line contains integers: the first number k_1 ($0 \leq k_1 \leq N$) — the number of monsters in the first row, the next k_1 numbers contain the numbers of cells (in ascending order) in which the monsters of the first row are located.

The third line contains integers: the first number k_2 ($0 \leq k_2 \leq N$) — the number of monsters in the second row, the next k_2 numbers contain the numbers of cells (in ascending order) in which the monsters of the second row are located.

Output

Output a single number — the smallest number of moves needed to clear the field of all monsters.

Note

Points for each subtask are awarded only if all tests for that subtask and the required subtasks are successfully completed. Some subtasks may also require that all tests in the problem statement are passed. For such subtasks the letter S is additionally indicated.

Subtask	Add. constraints	Points	Req. subtasks
1	$k_2 = 0$	15	S
2	$k_2 = 1$	15	1
3	$k_2 = 2$	15	2
4	None	55	3

Examples

standard input	standard output
5 3 1 4 5 1 1	2
7 3 1 2 7 4 2 3 4 7	3

E. Replacing characters

Time limit: 3 seconds
 Memory limit: 512 megabytes

You need to process q queries over a string s of length n . The type of a query is defined as follows:

1. In the substring $s_l s_{l+1} \dots s_{r-1} s_r$, all occurrences of the character x are replaced with the character y , and all occurrences of the character y are replaced with x .
2. Given a number i and a character x , the character x is inserted into the string after s_i . If $i = 0$, the character is inserted at the beginning of the string.
3. Given two numbers l, r , and a character x , one need to output the number of occurrences of x in the substring $s_l s_{l+1} \dots s_{r-1} s_r$.

Input

The first line contains an integer n ($1 \leq n \leq 5 \cdot 10^5$) – the initial length of the string.

The second line contains a string of length n consisting of lowercase Latin letters – the initial value of the string s .

The third line contains an integer q ($1 \leq q \leq 2 \cdot 10^5$) – the number of queries.

Each on next q lines contains description of queries.

Problem statements

- A query starts with an integer t ($1 \leq t \leq 3$) – its type.
- For the first type of query, the numbers l, r ($1 \leq l \leq r \leq |s|$) and the characters x, y are given – the boundaries of the substring and the characters to be replaced.
- For the second type of query – the number i ($0 \leq i \leq |s|$) and the character x are given - the position after which the character is inserted and the character to be inserted.
- For the third type of query – the numbers l, r ($1 \leq l \leq r \leq |s|$) and the character x are given - the boundaries of the substring and the character whose number of occurrences needs to be counted.

It is guaranteed that all characters are lowercase Latin letters.

Output

For each query of the third type, output the number of occurrences of a given character.

Example

standard input	standard output
7	2
abacaba	3
5	3
3 2 6 b	
3 3 7 a	
1 1 5 a b	
2 0 a	
3 1 8 a	

Note

The scores for each subtask are awarded if and only if all tests for that subtask and its required subtasks are successfully passed. For some subtasks, it may also be required to pass all tests from the problem statement. For such subtasks, the letter "S" is additionally indicated.

Subtask	Additional constraints	Score	Req. subtasks
1	$1 \leq n, q \leq 1000$	17	S
2	All letters can only be equal to a and b	13	—
3	All queries have $t = 3$	11	—
4	No queries have $t = 2$	19	3
5	None	40	S, 1, 2, 3, 4

F. The City View

Time limit: 1 second
Memory limit: 256 megabytes

The City is a checkered field of size $N \times M$. Each cell has a tower (possibly empty) of $1 \times 1 \times 1$ cubes. If you look at the City from the right, you can see the contours of N towers of height a_1, a_2, \dots, a_N (it turns out that a_i is the height of the highest tower in i -th row). If you look straight ahead, you can see the contours of M towers of height b_1, b_2, \dots, b_M (b_j is the height of the tallest tower in j -th column).

Write a program that uses these pictures to calculate the smallest possible number of cubes standing on the field.

Input

The first line contains two integers — N and M ($0 < N, M, N \cdot M \leq 10^6$).

The second line contains N space separated integers — a_i ($0 \leq a_i \leq 10^9$).

The third line contains M space separated integers — b_i ($0 \leq b_i \leq 10^9$).

Output

The program should output a single integer — the smallest possible number of cubes standing on the board.

Note

Points for the first subtask are awarded independently.

Points for subsequent subtasks are awarded only if all tests for this subtask and the required subtasks are successfully completed. Some subtasks

Problem statements

may also require that all tests from the problem statement be passed. For such subtasks the letter S is additionally specified.

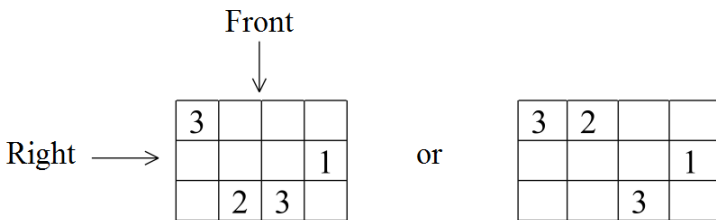
Subtask	Add. constraints	Points	Req. subtasks
1	$n, m \leq 30; a_i, b_j \leq 100$	22	—
2	$n, m, n \cdot m \leq 10^5; a_i, b_j \leq 1000$	15	S, 1
3	$a_i, b_j \leq 10^6$	15	2
4	None	48	3

Example

standard input	standard output
3 4 3 1 3 3 2 3 1	9

Note

Examples of the city view from above for a given test:



G. Magnets on the board

Time limit: 1 second
Memory limit: 256 megabytes

There are N magnets on the blackboard. Each magnet is numbered from 1 to N and has a string of text written on it, consisting of K lowercase latin

characters. You want to impress your informatics teacher by finding out the longest palindrome that is possible to compose with these magnets.

Input

The first line contains a 2 integers N, K ($1 \leq N \leq 10^5, 1 \leq N \cdot K \leq 10^6$) – the number of magnets on the board and length of each string.

Then N lines follow. The line i ($1 \leq i \leq N$) contains the string with length K written on the magnet i .

Output

Output a sequence of magnet numbers that form the longest palindrome.

It is guaranteed that answer exists. If there are multiple answers, print any of them.

Note

Points for each subtask are awarded only if all tests of this subtask and required subtasks are passed. Some subtasks may also require that all tests in the problem statement are passed. Such subtasks are indicated with the letter S.

Subtask	Add. constraints	Points	Req. subtasks
1	$K = 1$	10	–
2	$N \leq 5$	10	S
3	$N \leq 1000$	35	S, 2
4	None	45	S, 2, 3

Example

standard input	standard output
5 5 aabbb cccbb bbbaa bbccc xxxyy	1 2 4 3

Note

A palindrome is a string that reads the same from left to right and from right to left.

H. Schedule

Time limit: 1.5 seconds
Memory limit: 256 megabytes

On the planet *A* there are t days in a week. On this planet, there is a university *B* with a pretty interesting schedule. The students of the university either study all day or they have a day off. Also, all the students are very active: they attend all the cultural events that take place on their day off.

The schedule of all n events for the next week is out today. Events can last for several days. Each event schedule is represented as a string $a_1 \dots a_t$ of zeros and ones of length t , where $a_i = 1$ means that there is an event on i -th day of the week, and $a_i = 0$ means that there is no event on that day.

The university has a total of q groups, and their schedules look similar. That is, it is represented as a string $b_1 \dots b_t$ of zeros and ones of length t , where $b_i = 1$ means that the group studies all day on i -th day of the week, and $b_i = 0$ means that the group rests and students go to all the activities that take place that day.

An event is considered attended by a group if the group goes to all days of the event. Find out how many events each group will attend.

Input

First line contains three integers n , q and t – number of events, groups and days in a week ($1 \leq t \leq 20$, $1 \leq n, q \leq 10^5$).

Next n lines contain schedules of each event $a_1 \dots a_t$ ($a_i \in \{0, 1\}$).

Next q lines contain schedules of each group of students $b_1 \dots b_t$ ($b_i \in \{0, 1\}$).

Output

For each group of student output number of events visited by this group.

Note

The scores for each subtask are awarded if and only if all tests for that subtask and its required subtasks are successfully passed. For some subtasks, it may also be required to pass all tests from the problem statement. Such subtasks is additionally marked with letter "S".

Subtask	Add. constraints	Score	Req. subtasks
1	$n, q \leq 10^3$	15	S
2	Groups free days form a continuous segment	15	—
3	The days of the events form a continuous segment	15	—
4	None	55	S, 1, 2

Example

standard input	standard output
3 6 5	2
11011	0
01010	2
01110	3
00100	1
11001	0
10001	
00000	
10101	
01010	

I. Ring Road

Time limit: 1 second
 Memory limit: 256 megabytes

In a big city there are n squares connected by m two-way roads. In this city there can be no more than one road between any two squares. Also there is no road starting and finishing on the same square.

The mayor of this city decided to build a new road connecting two squares that would create **exactly one** new ring road consisting of the largest possible number of squares.

Problem statements

It is guaranteed that there is a way to construct a ring road satisfying such condition.

Help to find the amount of squares in the planned ring road.

Ring road is a closed chain of roads.

Input

The first line of the input data contains two integers n, m ($1 \leq n \leq 10^5, 1 \leq m \leq \min(\frac{n(n-1)}{2}, 10^7)$) – the number of squares and roads connecting them in this city.

Each of the next m lines contains two integers i, j ($1 \leq i, j \leq n$) – the roads connecting the squares with numbers i, j .

Output

Print the number of squares in the planned ring road as an output.

Building a new road must create one and only one new ring road.

Note

Each test is scored independently.

Subtask	Add. constraints	Points	Dependencies
1	There is no ring road in the city	30	–
2	None	70	–

Example

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

Note

There are three squares in the constructed ring road – 2, 3 and 4.

J. The Thought

Time limit: 3 seconds
 Memory limit: 1024 megabytes

Once upon a time UFO crashed in Tuymaada valley. Sakha scientists got a chance to study alien brain. Eventually, they discovered that the brain is structured as follows:

- There are m layers of n neurons in each.
- Every neuron from j -th layer ($0 \leq j \leq m-2$) is connected to some neuron from $(j+1)$ -th layer, let's define this connection as «edge». Neurons from layer $j+1$ can be connected to any number of neurons from layer j , but to only one neuron from layer $j+2$ if it exists.
- Each neuron has a weight a_i^j .

Scientists have also learned that the alien brain works in the following way:

- Signals (alien's thought) are generated in every neuron of layer $j = 0$ and have value $W_i^0 = 0$.
- Signal can propagate only from j -th layer to k -th, where $0 \leq j < k \leq m-1$, i.e. cannot propagate backwards.
- As the signal propagates along the edge from neuron with weight a_i^j on layer j to the neuron a_i^{j+1} on layer $j+1$, a signal with value $W_i^j + a_i^j \cdot a_i^{j+1}$ is propagated.
- The signal can also propagate from an arbitrary neuron a_i^j from the j -th layer to an arbitrary neuron a_i^k of k -th layer not along the edge (there may be an edge, but it does not have to go along it) with penalty $(k-j) \cdot a_i^j$, i.e. the value of the signal is increased by $a_i^j \cdot a_i^k - (k-j) \cdot a_i^j$, where $j < k$.
- A signal to the neuron can come only from one other neuron.

The scientists were surprised by the structure of alien brain and wondered what was the maximum value of a signal that could reach the last layer?

Input

First line contains two integers n, m ($1 \leq n \cdot m \leq 10^6$) – a number of neurons in a layer and a number of layers correspondingly.

Each of the next n lines contains $m - 1$ numbers e , where e on row i and column j means that i -th neuron on layer j is connected to e -th neuron on the layer $j + 1$ ($1 \leq e \leq n$).

Each of the next n lines contains m numbers that correspond to the weight of neuron a_i^j ($|a_i^j| \leq 10^5$).

Output

Your program should output the maximum value of a signal on the last layer.

Note

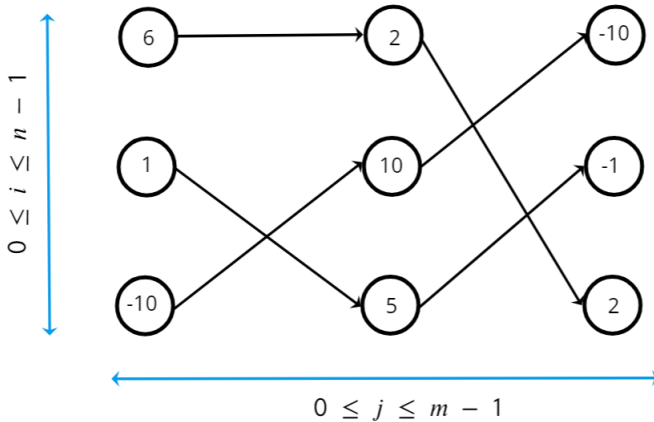
The scores for each subtask are awarded if and only if all tests for that subtask and its required subtasks are successfully passed. For some subtasks, it may also be required to pass all tests from the problem statement. Such subtasks is additionally marked with letter “S”.

Subtask	Add. constraints	Score	Req. Subtasks
1	$1 \leq n \cdot m \leq 10^3$	11	S
2	All a_i^j monotonically non-decreasing as j increases, $a_i^j \leq a_i^{j+1}$ for any i, l	7	–
3	All a_i^j monotonically non-increasing as j increases, $a_i^j \geq a_i^{j+1}$ for any i, l	15	–
4	$n = 1$	20	–
5	None	47	S, 1, 2, 3, 4

Examples

standard input	standard output
3 2 1 3 2 6 2 1 10 -2 5	54
3 3 1 3 3 1 2 2 6 2 -10 1 10 -1 -10 5 2	120

Note



The input of the second test from the statements is visualised here. The maximum signal is reached at the neuron (0,2) with signal value $W_0^2 = W_2^0 + a_2^0 \cdot a_0^2 - 2a_2^0 = 0 + 100 + 20 = 120$.

Solutions

A. Golfilliard

Under physical assumptions, it follows from the problem statement that the yellow ball after the impact will continue to move in the direction of the vector formed by the centers of the two balls. We need to find the corner where we need to send the white ball. To do this, imagine the final position of the white ball centered at a point that will lie on the line formed by the center of the yellow ball and the center of the hole.

You can calculate the desired position of the white ball at the moment of impact using the following chain of formulas:

1. Get a vector in the direction from the hole to the yellow ball

$$\vec{a} = (X_2 - X_3, Y_2 - Y_3)$$

2. Convert the resulting vector to a unit vector:

$$\vec{c} = \frac{\vec{a}}{\sqrt{X_a^2 + Y_a^2}}$$

3. Set aside two such unit vectors from the center of the yellow ball and get the desired position of the white ball at the moment of impact

$$Q = (X_2 + 2 * X_c, Y_2 + 2 * Y_c)$$

Now we need to find the angle between the x-axis and the vector, the direction of which is formed by the center of the initial position of the white ball and the point Q . To do this, you can use the formula:

$$\alpha = \arctan \frac{Y_Q - Y_1}{X_Q - X_1}$$

B. The librarian-robot

The problem can be solved using dynamic programming.

Let $dp[l][r]$ – the minimal number of operations, in which it is possible to remove all books from the interval $[l, r]$. Obviously, for any l $dp[l][l] = 1$ is satisfied.

For $l < r$:

- If $a_l = a_{l+1}$, then book l can always be removed along with book $l + 1$ (mentally «attach» book l to $l + 1$):

$$dp[l][r] = \min(dp[l][r], dp[l + 1][r]).$$

- If $a_r = a_{r-1}$, then similarly:

$$dp[l][r] = \min(dp[l][r], dp[l][r-1]).$$
- If $a_l = a_r$, then we can first clear everything in the interval $[l+1, r-1]$, and then get rid of l and r with one operation:

$$dp[l][r] = \min(dp[l][r], dp[l+1][r-1] + 1).$$
- In any other case we can split $[l, r]$ into two parts $[l, x]$ and $[x+1, r]$ such that:
 - $a_l = a_x \Rightarrow dp[l][r] = \min(dp[l][r], dp[l+1][x-1] + dp[x][r])$, which means we first clear everything in the interval $[l+1, x-1]$ and then attach book l to book x .
 - $a_r = a_x \Rightarrow dp[l][r] = \min(dp[l][r], dp[l][x] + dp[x+1][r-1])$, which means we first clear everything in the interval $[x+1, r-1]$ and then attach book r to book x .

The answer to the problem is $dp[1][n]$.

C. I'm the Map

The author's solution applies the binary lifting method to a function of one variable.

First, we need to find the value of a composite function of the form $f(g(x))$ over the whole definition area, i.e. for each vertex. If k is odd, the value of $f(x)$ is additionally counted separately. This is done in $O(n)$.

Next, a binary lift is done. Let $up[i][j]$ be the value of the chain of functions from i -th vertex, where the composite function $f(g(x))$ is repeated 2^j times. There is the following relation: $up[i][j] = up[up[i][j-1]][j-1]$. Such preprocessing will take $O(n \cdot \log_2 k)$.

Then k is divided in half, since $f(g(x))$ is calculated for one iteration.

Then in $O(\log_2 k)$ the answer to the chain of functions is found as follows: k is presented as the sum of powers of two, and for each power of two from lowest to highest the value of the chain of functions from the current value of x is determined.

For example, let $k = 6$. As a sum of powers of two, this is $4 + 2$, so one will need to make two iterations $x = up[x][1]$, $x = up[x][2]$ and get the answer.

Knowing the parent of each vertex, it remains to reconstruct the path from the library to the root.

D. A game for the gamer-student

In the first subtask you only need to cover the first row with the least number of 1×3 blocks so that all monsters are covered. This can be done greedily in one pass: set aside a block from the leftmost monster, then set aside a block from the leftmost unkilld monster, etc.

The second and third subtasks have one or two monsters each in the second row – you can carefully consider the monster/monsters of the second row and optimally cover them with blocks. And then greedily cover the first row.

To solve the complete problem, we need to apply dynamic programming.

Let $dp[i][x]$ – be the minimum number of blocks to cover all the monsters living in the first i left vertical rows AND in the profile x – we can code the profiles like this:

1. $x = 0$ – empty profile,
2. $x = 1$ – profile with one cell on top,
3. $x = 2$ – profile with two cells on top,
4. $x = 3$ – profile of one cell from the bottom,
5. $x = 4$ – profile of two cells from the bottom.

Next you need to correctly make the transition from $dp[i - 1][x]$ to $dp[i][y]$, depending on which cells the monsters are in.

The answer to the problem is $dp[n][0]$.

E. Replacing characters

To solve the first subtask, it is sufficient to store the string explicitly. For queries of the first and third types loop through the elements of substring and for queries of the second type insert a character in the right position.

For the third subtask for each letter we pre-calculate the number of its occurrences on each prefix. Then the answers to the queries will be $p[x][r] - p[x][l - 1]$, where $p[i][j]$ – the number of occurrences of character i in the prefix of string $s_1 \dots s_j$.

To solve the second subtask, we can construct an implicit Cartesian tree on the string. In each node, store the number a and b and a flag indicating whether or not a and b were substituted for each other in the subtree. And we need to use pending operations to query for character substitutions.

For a complete solution, we need to maintain a Cartesian tree for each letter by an explicit key on the string indices where the corresponding letters stand.

For queries of the first type, if $x \neq y$, we will use two *split* operations to extract a subtree with indices from l to r in each of the Cartesian trees corresponding to symbols x and y . Applying *merge*, we connect the tree with indices x from l to r with trees with indices from 1 to $l - 1$ and from $r + 1$ to $|s|$, obtained by *split* from the tree corresponding to y . Do the same with the tree y .

For queries of the second type, for all trees we extract a tree with indices $> i$. And add 1 to all its elements, using pending operations, since all character indices to the right of the inserted symbol will be shifted by 1. After that, insert an element $i + 1$ into the x tree.

For queries of the third type, we will simply extract a subtree with the desired indices and output its size. And join it back into one tree.

F. The City View

Consider an arbitrary value x : let $a_i = x$ for p values i , and $b_j = x$ for q values j .

In order to minimize the number of cubes, we put towers of height x only at the intersections of these rows and columns. It is necessary that for each row and each column of this sample there should be at least one tower. Hence, the smallest possible number of towers of height x will be $\min(p, q)$, and cubes – $x \cdot \min(p, q)$.

It remains to sum the number of cubes over all possible values of x .

To solve the first three subtasks, one can use a regular array to count the values of p and q for all x ($x \leq 10^6$).

For the fourth subtask, one can either do coordinate compression or use a data structure like `unordered_map` in C++.

G. Magnets on the board

We will consider the string y to be the reverse of string x if y can be obtained from x by placing the letters in reverse order. Consider the example of a palindrome $s_1s_2s_3 \dots s_{l-2}s_{l-1}s_l$, where s_i ($1 \leq i \leq l$) — strings of the same length. From the fact that the string is a palindrome, it follows that s_i is the reverse of s_{l-i+1} for any i from 1 to l .

Based on that, if the input contains a pair of strings s and $s_{reverse}$, then they can be part of a palindrome by placing them in the opposite sides of the resulting string — $s \dots s_{reverse}$. By splitting the strings from the input into such pairs, you can construct the maximum palindrome. It is also worth considering that in the input strings, there may be those that are initially palindromes. And if we made up a palindrome from an even number of strings, but we still have an unused palindrome input string, then the answer can be improved by placing it in the middle of the received answer.

Searching for s and $s_{reverse}$ pairs can be done efficiently using a hash table data structure.

H. Schedule

For the first subtask, it is sufficient to bruteforce for $\mathcal{O}(nqt)$.

To solve the second and third subtasks, for each group you need to maintain its left and right boundary in the two-dimensional array $t \times t$ and update it like this:

```
void Update(
    std::vector<std::vector<int>>& a, int x, int y
) {
    for (int i = x; i >= 0; --i) {
        for (int j = y; j < a[x].size(); ++j) {
            ++a[i][j];
        }
    }
}
```

And then for each event also find its left and right boundary and output $a[i][r]$; if group's free days do not specify a continuous segment, then you can update all its sub-segments.

For a complete solution, it is sufficient to calculate the sum of the indicators over all submasks of the mask:

```
for (int i = 0; i < t; ++i) {
    for (int mask = 0; mask < (1 << t); ++mask) {
        if (!(mask & (1 << i))) {
            events[mask + (1 << i)] += events[mask];
        }
    }
}
```

I. Ring Road

One needs to construct a graph of bridges for the given graph. Obtained graph will be made of trees. Let's find the maximum diameter from that trees. The answer will be greater than found maximum by 1.

Indeed, let there be a path between vertices i and j that doesn't contain a bridge. Hence, adding an edge $\langle i, j \rangle$ will result in one more cycle. And if the graph has only bridges, you can connect any vertices and get only one cycle.

J. The Thought

To solve the first subtask it is sufficient to carefully brute force in $O(n^2m^2)$.

When weights of neurons increase, one can use greedy algorithm and consider only the previous layer and the maximum negative value on the first layer where negative values appear, if any.

Next, let's note that:

$$W_i^k = W_i^j + a_i^j a_i^k - (k - j) a_i^j = W_i^j + a_i^j a_i^k - k a_i^j + j a_i^j = (a_i^j, W_i^j + j a_i^j) \cdot (a_i^k - k, 1).$$

For transition from j -th layer to k -th, one can find the maximum scalar product from previous vectors of type $(a_i^j, W_i^j + j a_i^j)$ and vector $(a_i^k - k, 1)$. This can be done by maintaining a convex hull at such points. In the third subtask it is enough to make it on stack and remove last points, since a_i^j are monotonically non-increasing.

For a complete solution one needs to use a search tree in order to calculate a scalar product and use top-down traversal. I.e. separately maintain upper

and lower hulls, for the leftmost point find scalar product and then use the binary search to find maximum value. To update the hull one can use dynamic convex hull – check if there the point above or below the hull that is needed to be inserted. And then insert it removing some points left and right from it.

It can be tedious to implement this but one can turn $y = a_i x + (W_i + j a_i)$ to linear functions and use Li Chao tree for finding maximum product.

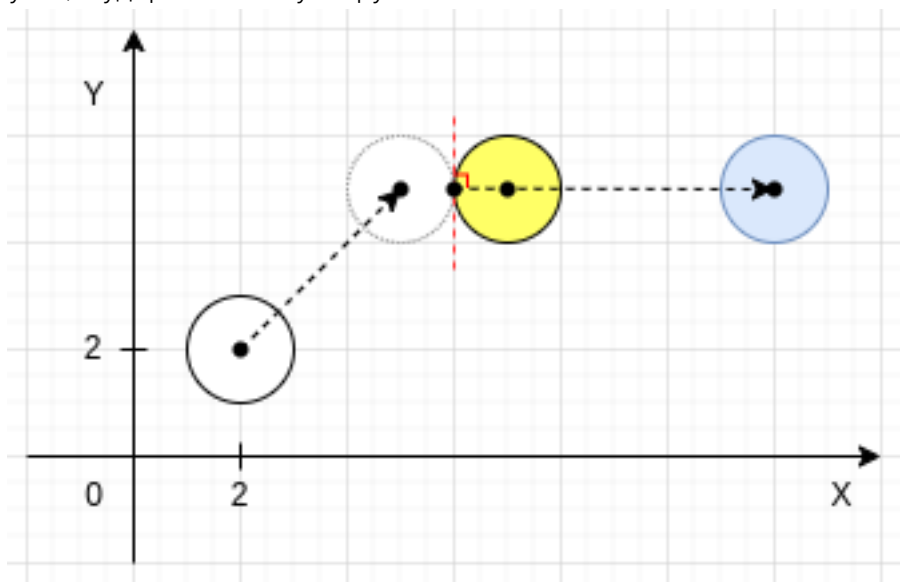
Задачи

А. Гольфильярд

Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Программист Петя после разгромного проигрыша в гольфильярд решил создать программу, которая смогла бы вычислить, в каком направлении надо ударить по белому шару, чтобы цветной шар попал точно в центр лунки.

Конечно, в его виртуальном гольфильярде трение, как и сопротивление воздуха, отсутствуют полностью, столкновение шаров абсолютно упругое, и удар по белому шару является плоским.



Формат входных данных

В первой строке заданы координаты центра (X_1, Y_1) белого шара. Во второй строке — координаты центра (X_2, Y_2) цветного шара. В третьей строке — координаты (X_3, Y_3) центра лузы. Все числа целые и лежат в диапазоне от -1000 до 1000.

Радиусы шаров одинаковы и равны 1.

Формат выходных данных

Вывести угол α в градусах относительно оси Ox против часовой стрелки, в направлении которого необходимо ударить белый шар ($0 \leq \alpha < 360$). Ответ вывести с точностью не хуже 4-х знаков после запятой.

Гарантируется, что ответ существует.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
2 2 7 5 12 5	45.0000

Замечание

При плоском ударе шар, по которому бьют, до соударения с прицельным шаром скользит без какого-либо вращения.

В. Робот-библиотекарь

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В библиотеку для переноса книг с полки на полку приняли на работу робота, снабжённого искусственным интеллектом. Он может выполнять только одно действие: снять любое количество стоящих подряд книг с полки A и переместить их на другую полку (кроме полки A в библиотеке имеется еще K полок, все эти полки пронумерованы числами от 1 до K). Все оставшиеся книги на полке A после этого уплотняются, то есть сдвигаются в сторону левого конца полки.

Накануне робот хорошо поработал и полка A опустела. Но сегодня в библиотеку привезли очередную партию книг и поместили их на полку A . Каждая книга помечена числом, обозначающим, на какую полку она должна быть перемещена. Напишите программу, рассчитывающую наименьшее количество действий, чтобы переместить все книги с полки A на свои полки.

Формат входных данных

Первая строка содержит число $1 \leq N \leq 200$ — количество привезённых книг.

Вторая строка содержит число $1 \leq K \leq 200$ — количество различных полок в библиотеке (кроме полки А).

В третьей строке заданы числа a_1, a_2, \dots, a_N — номера полок, на которые должны быть перенесены книги, установленные на полку А.

Формат выходных данных

Выведите единственное число — наименьшее количество действий робота.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Доп. ограничения	Баллы	Необх. подзадачи
1	$K \leq 1$	7	—
2	$K \leq 2, N \leq 20$	8	1
3	$K \leq 2$	14	1, 2
4	$N, K \leq 10$	6	—
5	$K \leq 10$	25	1, 2, 3, 4
6	Нет	40	1, 2, 3, 4, 5

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
6 3 2 1 1 3 1 2	3

Замечание

В примере сначала робот ставит 4-ю книгу на полку 3. Затем книги с индексами 2, 3, 4 на полку 1. Затем книги с индексами 1, 2 на полку 2.

С. Я — Карта

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Даше-путешественнице нужно успеть отнести прочитанные книги в библиотеку до её закрытия. Но вот незадача, из того места, куда привело её путешествие, она не знает самого короткого пути к библиотеке. Даше нужна ваша помощь. В её рюкзаке имеется карта. Скажите «Карта!» и определите кратчайший путь к библиотеке.

Карта собой представляет неориентированное дерево, где корнем дерева является то место, где сейчас находится Даша. Каждая вершина пронумерована от 0 до $n - 1$. Заданы две функции:

1. Функция $f(x)$, которая принимает номер вершины и возвращает номер вершины, равный $x \oplus p[x]$ по модулю размера дерева, где $p[x]$ — родитель x . Считается, что родителем корня является сам корень.
2. Функция $g(x)$, которая принимает номер вершины и возвращает номер вершины, равный $x \mid p[x]$ по модулю размера дерева.

Адрес библиотеки вычисляется следующим образом: $f(g(f(g \dots (x) \dots))$. Функции чередуются и суммарно выполняются k -раз. Цепочка функций начинается с функции f (то есть самой внешней функцией является функция f).

Помогите Даше найти библиотеку и построить кратчайший маршрут до неё.

Формат входных данных

В первой строке заданы три числа n, k, s — соответственно, количество вершин дерева, количество функций в цепочке и номер вершины, где находится Даша ($0 < n \leq 10^5, 0 < k \leq 10^{18}, 0 \leq s < n$).

В следующих $n - 1$ строках через пары номеров вершин a и b ($0 \leq a, b < n$) описаны ребра.

В последней строке задано одно число x — номер стартовой вершины цепочки функций ($0 \leq x < n$).

Формат выходных данных

В первой строке выведите «Map!» без кавычек.

Во второй строке выведите путь от той вершины, где находится Даша, до библиотеки. Номера вершин выводить через пробел.

Система оценивания

Баллы за каждый тест начисляются независимо.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
7 2 2 0 1 5 1 1 6 2 0 2 4 2 3 6	Map! 2
7 3 6 4 3 2 1 0 1 3 1 2 6 2 5 4	Map! 6 2 1 3

Замечание

1. $a \oplus b$ — это побитовая операция сложения по модулю 2 (XOR) чисел a и b .
2. $a | b$ — это побитовая операция логического сложения (OR) чисел a и b .
3. Дерево — это ациклический связанный граф.
4. Размер дерева — количество вершин в дерева.

D. Игра для студента-геймера

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Многие IT-студенты любят играть в компьютерные игры.

Преподаватель по программированию предложил такую игру: имеется поле $2 \times N$ (2 ряда по N клеток), в некоторых клетках которого находятся монстры. У вас есть только одна способность: за один ход сжечь дотла три клетки, идущие подряд; при этом монстры в этих клетках погибают. Получается, что за один ход вы можете сжечь либо вертикальный ряд из двух клеток, либо горизонтальный — из трёх (или меньше, если сжигать на краю поля).

Разумеется, преподаватель ожидает от вас написанную программу, которая за *наименьшее* количество ходов полностью очистит поле от монстров.

Формат входных данных

Первая строка содержит единственное число N ($0 < N \leq 3 \cdot 10^5$).

Вторая строка содержит целые числа: первое число k_1 ($0 \leq k_1 \leq N$) — количество монстров в первом ряду, следующие k_1 чисел содержат номера клеток (в порядке возрастания), в которых находятся монстры первого ряда.

Третья строка содержит целые числа: первое число k_2 ($0 \leq k_2 \leq N$) — количество монстров во втором ряду, следующие k_2 чисел содержат номера клеток (в порядке возрастания), в которых находятся монстры второго ряда.

Формат выходных данных

Вывести единственное число — наименьшее количество ходов, необходимое для очистки поля от всех монстров.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

Подзадача	Доп. ограничения	Баллы	Необх. подзадачи
1	$k_2 = 0$	15	У
2	$k_2 = 1$	15	1
3	$k_2 = 2$	15	2
4	Нет	55	3

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
5 3 1 4 5 1 1	2
7 3 1 2 7 4 2 3 4 7	3

Е. Замена символов

Ограничение по времени: 3 секунды

Ограничение по памяти: 512 мегабайт

Вам даны начальное значение строки s длины n и q запросов. Вам нужно обработать запросы трёх типов:

1. На подотрезке $s_l s_{l+1} \dots s_{r-1} s_r$ все символы x заменяются символами y , а все символы y заменяются x .
2. Дано число i и символ x . В строку после s_i вставляется символ x . Если $i = 0$, символ вставляется в начало строки.
3. Даны два числа l, r и символ x . Вам нужно вывести количество вхождений x в подотрезок $s_l s_{l+1} \dots s_{r-1} s_r$.

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 5 \cdot 10^5$) — начальная длина строки.

Во второй строке содержится строка из строчных латинских букв длины n — начальное значение строки s .

В третьей строке дано число q ($1 \leq q \leq 2 \cdot 10^5$) — количество запросов.

В следующих q строках содержится описание запросов:

- Каждый запрос начинается с целого числа t ($1 \leq t \leq 3$) — типа запроса.
- Для запроса первого типа далее даны числа l, r ($1 \leq l \leq r \leq |s|$) и символы x, y — соответственно, границы подотрезка и символы, которые нужно заменить;
- Для запроса второго типа — число i ($0 \leq i \leq |s|$) и символ x — соответственно, позиция, после которой вставляется символ, и вставляемый символ;
- Для запроса третьего типа — числа l, r ($1 \leq l \leq r \leq |s|$) и символ x — соответственно, границы подотрезка и символ, чьё количество вхождений нужно найти.

Гарантируется, что все символы — латинские строчные буквы.

Формат выходных данных

Для каждого запроса третьего типа выведите количество вхождений.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

Подзадача	Доп. ограничения	Баллы	Необх. подзадачи
1	$1 \leq n, q \leq 1000$	17	У
2	Все буквы могут быть равны только a и b	13	—
3	У всех запросов $t = 3$	11	—
4	Нет запросов $t = 2$	19	3
5	Нет доп. огр.	40	У, 1, 2, 3, 4

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
7	2
abacaba	3
5	3
3 2 6 b	
3 3 7 a	
1 1 5 a b	
2 0 a	
3 1 8 a	

Ф. Вид на город с разных сторон

Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Город — это клетчатое поле размера $N \times M$. На каждой клетке расположена башня (возможно, пустая) из кубиков $1 \times 1 \times 1$. Если смотреть на Город справа, то видны контуры N башен высоты a_1, a_2, \dots, a_N (получается, что a_i — высота самой высокой башни в i -ом ряду). Если смотреть прямо, то видны контуры M башен высоты b_1, b_2, \dots, b_M (b_j — высота самой высокой башни в j -й колонне).

Напишите программу, которая по этим картинкам вычисляет наименьшее возможное количество кубиков, стоящих на поле.

Формат входных данных

В первой строке заданы целые числа N и M ($0 < N, M, N \cdot M \leq 10^6$).

Во второй строке заданы N целых чисел a_i , разделенных пробелами ($0 \leq a_i \leq 10^9$).

В третьей строке заданы M целых чисел b_j , разделенных пробелами ($0 \leq b_j \leq 10^9$).

Формат выходных данных

Программа должна вывести одно целое число — наименьшее возможное количество кубиков, стоящих на поле.

Система оценивания

Баллы за первую подзадачу начисляются независимо.

Баллы за каждую последующие подзадачи начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

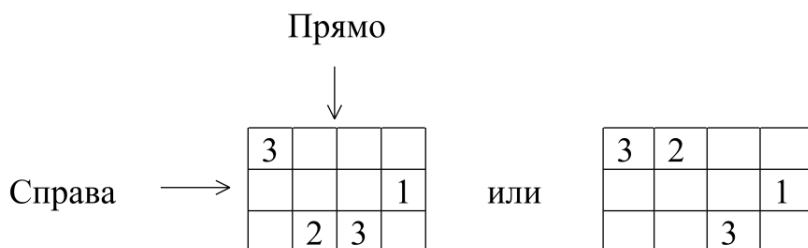
Подзадача	Доп. ограничения	Баллы	Необх. подзадачи
1	$n, m \leq 30$ $a_i, b_j \leq 100$	22	—
2	$n, m, n \cdot m \leq 10^5$ $a_i, b_j \leq 1000$	15	У, 1
3	$a_i, b_j \leq 10^6$	15	2
4	Нет	48	3

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
3 4 3 1 3 3 2 3 1	9

Замечание

Примеры вида на город сверху для приведенного примера:

**Г. Магнитики на доске**

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

На школьной доске N магнитиков. Каждый магнитик пронумерован числами от 1 до N и содержит строку, состоящую из K строчных латинских символов. Вы хотите впечатлить своего учителя информатики, найдя самый длинный палиндром, который можно составить из этих магнитиков.

Формат входных данных

Первая строка содержит 2 целых числа N, K ($1 \leq N \leq 10^5$, $1 \leq N \cdot K \leq 10^6$) — соответственно, количество магнитов на доске и количество символов в каждой строке.

Дальше следуют N строк. В строке i ($1 \leq i \leq N$) содержится строка длины K , написанная на магнитике i .

Формат выходных данных

Выведите последовательность чисел, соответствующую номерам магнитов, которые образуют длиннейший палиндром.

Гарантируется, что ответ существует. Если ответов несколько, то выведите любой из них.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Задачи

Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

Подзадача	Доп. ограничения	Баллы	Необх. подзадачи
1	$K = 1$	10	—
2	$N \leq 5$	10	У
3	$N \leq 1000$	35	У, 2
4	Нет	45	У, 2, 3

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
5 5 aabbb cccbb bbbaa bbccc xxxуу	1 2 4 3

Замечание

Палиндром — это строка, которая одинаково читается как слева направо, так и справа налево.

Н. Расписание

Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

На планете А одна неделя состоит из t суток. На этой планете есть университет В с весьма оригинальным учебным расписанием. Студенты университета либо учатся весь день, либо отдыхают. А еще все студенты университета очень активные, они посещают все культурные мероприятия, которые проходят в их свободный день.

Сегодня вышло расписание всех n мероприятий на всю следующую неделю. Мероприятия могут идти несколько дней. Расписание каждого

мероприятия представлено в виде строчки $a_1 \dots a_t$ из нулей и единичек длины t , где $a_i = 1$ означает, что мероприятие проходит в i -й день недели, а $a_i = 0$ означает, что его в этот день нет.

Всего в университете учатся q групп, и их расписание выглядит аналогично. То есть оно представлено в виде строчки $b_1 \dots b_t$ из нулей и единичек длины t , где $b_i = 1$ означает, что группа учится весь день в i -й день недели, а $b_i = 0$ означает, что группа отдыхает и студенты идут на все мероприятия, которые проходят в этот день.

Мероприятие считается посещенным группой, если группа сходила на все дни его проведения. Выясните, сколько мероприятий посетит каждая группа на следующей неделе.

Формат входных данных

В первой строке заданы три натуральных числа n , q и t — количество мероприятий, количество групп и количество дней в неделе соответственно ($1 \leq t \leq 20$, $1 \leq n, q \leq 10^5$).

В следующих n строках заданы расписания мероприятий $a_1 \dots a_t$ ($a_i \in \{0, 1\}$).

Затем следуют q строк, в каждой из которых заданы расписания групп $b_1 \dots b_t$ ($b_i \in \{0, 1\}$).

Формат выходных данных

Для каждой группы выведите количество мероприятий на следующей неделе, которые она посетит.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

Подз.	Доп. ограничения	Баллы	Необх. подзадачи
1	$n, q \leq 10^3$	15	У
2	Свободные дни у групп образуют непрерывный отрезок	15	—
3	Дни проведения мероприятий образуют непрерывный отрезок	15	—
4	Нет	55	У, 1, 2

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
3 6 5	2
11011	0
01010	2
01110	3
00100	1
11001	0
10001	
00000	
10101	
01010	

I. Кольцевая дорога

Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

В городе n площадей, связанных между собой m дорогами с двухсторонним движением. В этом городе между любыми двумя площадями не может быть более одной дороги. Также отсутствуют дороги которые начинаются и заканчиваются на одной и той же площади.

Мэр города решил соединить две площади двухсторонней дорогой так, чтобы появилась **ровно одна** новая кольцевая дорога с наибольшим количеством площадей.

Гарантируется, что кольцевую дорогу с заданным условием можно построить. Помогите найти количество площадей в проектируемой кольцевой дороге.

Кольцевая дорога — это замкнутая цепочка дорог.

Формат входных данных

В первой строке заданы два целых числа n , m ($1 \leq n \leq 10^5, 1 \leq m \leq \min(\frac{n(n-1)}{2}, 10^7)$) — соответственно, число площадей в городе и дорог, соединяющих эти площади.

В следующих m строках заданы по два целых числа i, j ($1 \leq i, j \leq n$) — дороги, соединяющие площади с номерами i, j .

Формат выходных данных

В единственную строку выведите искомое число площадей в проектируемой кольцевой дороге.

Система оценивания

Баллы за каждый тест в подзадаче начисляется независимо.

Подзадача	Ограничения	Баллы	Необх. подзадачи
1	В городе отсутствуют кольцевые дороги	30	—
2	Нет	70	—

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
5 5 1 3 1 5 2 3 2 4 3 5	3

Замечание

В построенной кольцевой дороге три площади — 2, 3 и 4.

В результате строительства новой дороги должна появиться одна и только одна новая кольцевая дорога.

Ж. Мысль

Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

На долину Туймаада упала летающая тарелка с инопланетянами. У якутских ученых появилась возможность исследовать их мозг.

Оказалось, что их мозг имеет следующую структуру:

- Мозг содержит m слоев по n нейронов.
- Каждый нейрон из j -го слоя ($0 \leq j \leq m-2$) связан с каким-то одним нейроном из $(j+1)$ -го слоя, назовем эту связь «ребром». Каждый нейрон из $(j+1)$ -го слоя может быть связан с каким угодно количеством нейронов из j -го слоя, но только с одним из $(j+2)$ -го слоя, если он существует.
- Каждый нейрон имеет некоторый вес a_j^i .

Специалисты также выяснили, что мозг инопланетян работает следующим образом:

- Сигналы (мысль инопланетянина) рождаются на каждом нейроне первого слоя ($j=0$) и имеют мощности $W_i^0 = 0$.
- Сигнал может перемещаться только из j -го слоя к k -му, где $0 \leq j < k \leq m-1$, то есть не может идти обратно.
- При перемещении сигнала по рёбрам из нейрона весом a_j^i на j -м слое к нейрону весом a_i^{j+1} на $(j+1)$ -м слое, передается сигнал с мощностью $W_i^j + a_j^i \cdot a_i^{j+1}$.
- Сигнал также может перемещаться из произвольного нейрона a_j^i из j -го слоя к произвольному нейрону a_i^k из k -го слоя, где $j < k$, не по рёбрам (может быть ребро и есть, но не обязательно идти по нему) со штрафом $(k-j) \cdot a_j^i$. То есть сигнал увеличивается на $a_j^i \cdot a_i^k - (k-j) \cdot a_j^i$.
- В нейрон может переместиться сигнал только из одного нейрона.

Ученые удивились такому строению мозга инопланетянина и решили посчитать, сигнал какой максимальной мощности может достичь последнего слоя?

Формат входных данных

Первая строка содержит два натуральных числа n, m ($1 \leq n \cdot m \leq 10^6$) — количество нейронов на каждом слое и количество слоев соответственно.

Далее следуют n строк по $m-1$ чисел e , где e в i -й строке и j -м столбце означает, что нейрон в i -й строке и j -м слое связан с e -м нейроном на $(j+1)$ -м слое ($1 \leq e \leq n$).

Затем следуют n строк по m чисел, задающие веса нейронов a_i^j ($|a_i^j| \leq 10^5$).

Формат выходных данных

Выведите ответ на задачу.

Система оценивания

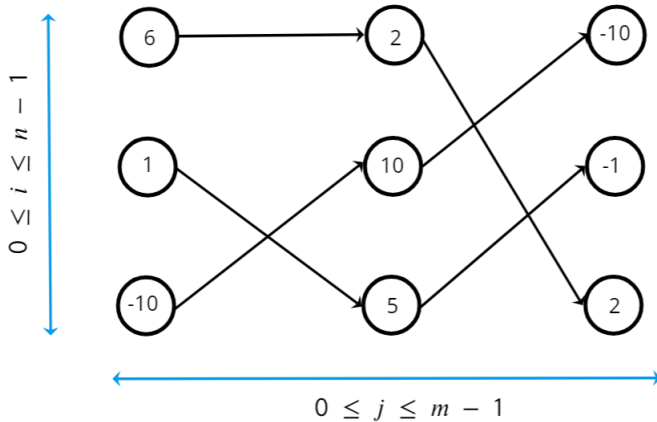
Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

Подз.	Доп. ограничения	Баллы	Необх. под.
1	$1 \leq n \cdot m \leq 10^3$	11	У
2	Все a_i^j монотонно неубывают с ростом j , $a_i^j \leq a_i^{j+1}$ для любых i, l	7	—
3	Все a_i^j монотонно невозрастают с ростом j , $a_i^j \geq a_i^{j+1}$ для любых i, l	15	—
4	$n = 1$	20	—
5	Нет	47	У, 1, 2, 3, 4

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
3 2 1 3 2 6 2 1 10 -2 5	54
3 3 1 3 3 1 2 2 6 2 -10 1 10 -1 -10 5 2	120

Замечание



Здесь визуализирован вход второго теста из условия. Максимальный сигнал достигается в вершинке $(0, 2)$ мощностью $W_0^2 = W_2^0 + a_2^0 \cdot a_0^2 - 2a_2^0 = 0 + 100 + 20 = 120$.

Решения

А. Гольфийярд

При физических допущениях из условия задачи цветной шар после удара белым шаром будет двигаться вдоль прямой, соединяющей центры шаров в момент их касания.

Нам нужно найти угол, в направлении которого необходимо ударить белый шар. Для этого представим себе конечное положение белого шара с центром в точке, которая будет лежать на прямой, образованной центром цветного шара и центром лунки.

Вычислить нужное положение белого шара в момент касания цветного можно по следующим формулам:

1. Получить вектор в направлении от центра лунки к центру цветного шара:

$$\vec{a} = \{X_2 - X_3; Y_2 - Y_3\}.$$

2. Преобразовать полученный вектор в единичный:

$$\vec{c} = \frac{\vec{a}}{\sqrt{X_a^2 + Y_a^2}}.$$

3. Отложить от центра цветного шара два таких единичных вектора и получить координаты белого шара в момент удара по цветному:

$$Q = (X_2 + 2 * X_c; Y_2 + 2 * Y_c).$$

Теперь нужно найти угол между осью абсцисс и вектором, направление которого образуется центром начального положения белого шара и точкой Q . Для этого можно воспользоваться формулой:

$$\alpha = \arctan\left(\frac{Y_Q - Y_1}{X_Q - X_1}\right).$$

В. Робот-библиотекарь

Задача решается методом динамического программирования.

Пусть $dp[l][r]$ — минимальное количество операций, за которое можно убрать все книги из подотрезка $[l, r]$. Очевидно, что для любого l выполнено $dp[l][l] = 1$.

Для $l < r$:

- Если $a_l = a_{l+1}$, то книгу l всегда можно убрать вместе с книгой $l + 1$, мысленно «прикрепим» книгу l к $l + 1$:

$$dp[l][r] = \min(dp[l][r], dp[l + 1][r]).$$

- Если $a_r = a_{r-1}$, то аналогично:

$$dp[l][r] = \min(dp[l][r], dp[l][r - 1]).$$

- Если $a_l = a_r$, то можно сначала очистить всё в интервале $[l + 1, r - 1]$, и затем одной операцией избавиться от l и r .

$$dp[l][r] = \min(dp[l][r], dp[l + 1][r - 1] + 1).$$

- В любом другом случае можно разбить $[l, r]$ на две части $[l, x]$ и $[x + 1, r]$ таких, что:

- $a_l = a_x \Rightarrow dp[l][r] = \min(dp[l][r], dp[l + 1][x - 1] + dp[x][r])$, то есть мы сначала очищаем всё в интервале $[l + 1, x - 1]$, а затем прикрепляет книгу l к книге x .

- $a_r = a_x \Rightarrow dp[l][r] = \min(dp[l][r], dp[l][x] + dp[x + 1][r - 1])$, то есть мы сначала очищаем всё в интервале $[x + 1, r - 1]$, а затем прикрепляет книгу r к книге x .

Ответ на задачу: $dp[1][n]$.

С. Я — Карта

Авторское решение применяет метод бинарного подъёма для функции от одной переменной.

Сначала требуется найти значение сложной функции вида $f(g(x))$ на всей области определения, то есть для каждой вершины. При нечётном k дополнительно отдельно считается значение $f(x)$. Делается это за $O(n)$.

Далее делается бинарный подъём. Пусть $up[i][j]$ — значение цепочки функций от i -й вершины, где сложная функция $f(g(x))$ повторяется 2^j раз. Имеется следующая зависимость: $up[i][j] = up[up[i][j - 1]][j - 1]$. Такой препроцессинг займет $O(n \cdot \log_2 k)$.

Затем k делится пополам, поскольку $f(g(x))$ считается за одну итерацию.

Затем за $O(\log_2 k)$ ищется ответ на цепочку функций следующим образом: k представляется в виде суммы степеней двоек, и для каждой степени двойки от младшей к старшей определяется значение цепочки функций от текущего значения x .

Пусть, например, $k = 6$. В виде суммы степеней двоек это $4 + 2$, поэтому нужно будет сделать две итерации $x = up[x][1]$, $x = up[x][2]$ и получить ответ.

Остается, зная родителя каждой вершины, восстановить путь от библиотеки до корня.

D. Игра для студента-геймера

В первой подзадаче нужно только покрыть первый ряд наименьшим числом блоков 1×3 так, чтобы все монстры покрылись. Это можно сделать жадно за один проход: отложим блок от самого левого монстра, далее отложим блок от самого левого неубитого монстра, и т. д.

Во второй и третьей подзадачах имеется по одному или двум монстрам во втором ряду — можно аккуратно рассмотреть монстра/монстров второго ряда и оптимально покрыть их блоками. А далее жадно докрыть первый ряд.

Для решения полной задачи нужно применить динамическое программирование (можно сказать, по профилю).

Пусть $dp[i][x]$ — это минимальное количество блоков, чтобы покрыть всех монстров, находящихся в первых i слева вертикальных рядах и в профиле x , который можно закодировать так:

1. $x = 0$ — пустой профиль,
2. $x = 1$ — профиль из одной клетки сверху на $(i + 1)$ -й позиции,
3. $x = 2$ — профиль из двух клеток сверху на $(i + 1)$ -й и $(i + 2)$ -й позициях,
4. $x = 3$ — профиль из одной клетки снизу на $(i + 1)$ -й позиции,
5. $x = 4$ — профиль из двух клеток снизу на $(i + 1)$ -й и $(i + 2)$ -й позициях.

Далее нужно правильно сделать переход с $dp[i - 1][]$ на $dp[i][]$, в зависимости от того, в каких клетках находятся монстры.

Ответ на задачу: $dp[n][0]$.

Е. Замена символов

Для решения первой подзадачи достаточно явно хранить строку. Для запросов первого и третьего типов циклом пройтись по элементам подстроки и для запросов второго типа вставлять символ в нужную позицию.

Для третьей подзадачи для каждой буквы предварительно подсчитаем количество вхождений этой буквы на каждом префиксе. Тогда ответы на запросы будут равны $p[x][r] - p[x][l - 1]$, где $p[i][j]$ — количество вхождений символа i в префиксе строки $s_1 \dots s_j$.

Для решения второй подзадачи можно построить неявное декартово дерево на строке. В каждой вершине хранить количество a и b и флаг, указывающий на то, заменялись ли в поддереве друг на друга символы a и b . И нужно использовать отложенные операции для запроса на замену символов.

Для полного решения будем поддерживать для каждой буквы декартово дерево по явному ключу на индексах строки, в которых стоят соответствующие буквы.

Для запросов первого типа, если $x \neq y$, с помощью двух операций *split* будем вычленять в декартовых деревьях, соответствующих символам x и y , по одному отдельному поддереву с индексами от l до r . Применяв *merge*, соединим дерево с индексами вхождений x от l до r с деревьями с индексами от 1 до $l - 1$ и от $r + 1$ до $|s|$, получившимися при *split* от дерева, соответствующего y . Аналогично поступим с деревом y .

Для запросов второго типа для всех деревьев вычленим дерево с индексами $> i$. И ко всем её элементам, с помощью отложенных операций, прибавим 1 , так как все индексы символов правее вставленного символа сместятся на 1 . После этого вставим в дерево x элемент $i + 1$.

Для запросов третьего типа будем просто вычленять поддерево с нужными индексами, выводить его размер. И соединять обратно в одно дерево.

Ф. Вид на город с разных сторон

Рассмотрим произвольное значение x : пусть $a_i = x$ для p значений i , и $b_j = x$ для q значений j .

С целью минимизации количества кубиков поставим башни высотой x только в пересечениях этих строк и столбцов. При этом нужно, чтобы для каждой строки и каждого столбца этой выборки была хотя бы одна башня. Следовательно, наименьшее возможное количество башен высоты x будет равно $\min(p, q)$, а кубиков — $x \cdot \min(p, q)$.

Остается просуммировать количество кубиков по всем возможным значениям x .

Для решения первых трёх подзадач можно воспользоваться обычным массивом, чтобы подсчитывать значения p и q для всех x ($x \leq 10^6$).

Для четвёртой подзадачи можно либо сделать сжатие координат, либо воспользоваться структурой данных типа `unordered_map` в C++.

Г. Магнитики на доске

Будем считать строку u обратной к строке x , если u можно получить из x , разместив буквы в обратном порядке. Рассмотрим пример палиндрома $s_1 s_2 s_3 \dots s_{l-2} s_{l-1} s_l$, где s_i ($1 \leq i \leq l$) — строки одинаковой длины. Из определения палиндрома следует, что s_i — строка, обратная к s_{l-i+1} для любого i от 1 до l .

Исходя из этого, если во входных данных есть пара из строк s и $s_{\text{обратная}}$, то они могут стать частью палиндрома, если разместить их на противоположных сторонах результирующей строки — $s \dots s_{\text{обратная}}$. Разбив строки из входных данных на такие пары, можно составить максимальный палиндром. Стоит также учесть, что во входных строках, могут быть те, которые изначально являются палиндромами. И, если мы составили палиндром из четного числа строк, но у нас остается неиспользованная строка-палиндром, то ответ можно улучшить, разместив эту строку в середину полученного ответа.

Поиск пар s и $s_{\text{обратная}}$ можно эффективно выполнить, используя структуру данных «хеш-таблица».

Н. Расписание

Для первой подзадачи достаточно написать полный перебор за $O(nqt)$.

Чтобы решить вторую и третью подзадачу, нужно для каждой группы поддерживать его левую и правую границу в двумерном массиве $t \times t$ и обновлять его так:

```
void Update(
    std::vector<std::vector<int>>& a, int x, int y
) {
    for (int i = x; i >= 0; --i) {
        for (int j = y; j < a[x].size(); ++j) {
            ++a[i][j];
        }
    }
}
```

А затем для каждого мероприятия тоже находить его левую и правую границу и выводить $a[l][r]$. Если же свободные дни группы не задают непрерывный отрезок, то можно обновить все его подотрезки.

Для полного решения достаточно посчитать сумму индикаторов по всем подмаскам маски:

```
for (int i = 0; i < t; ++i) {
    for (int mask = 0; mask < (1 << t); ++mask) {
        if (!(mask & (1 << i))) {
            events[mask + (1 << i)] += events[mask];
        }
    }
}
```

I. Кольцевая дорога

Построить граф из мостов заданного графа. Построенный граф будет состоять из деревьев. Найдём максимальный диаметр полученных деревьев. Ответ будет на единицу больше.

Действительно, пусть между вершинами i и j имеется путь, не содержащий мост. Тогда при добавлении ребра $\langle i, j \rangle$ появится еще один

цикл. А если в графе одни мосты, то можно соединять любые вершины и получать только один цикл.

Ж. Мысль

Для первой подзадачи достаточно правильно понять условие и написать полный перебор за $O(n^2m^2)$.

Когда веса в нейронах возрастают, можно написать жадный алгоритм и следить только за предыдущим слоем и максимальным отрицательным значением на первом слое, где он встретился, если он есть.

Далее заметим следующую вещь:

$$W_i^k = W_i^j + a_i^j a_i^k - (k - j) a_i^j = W_i^j + a_i^j a_i^k - k a_i^j + j a_i^j = (a_i^j, W_i^j + j a_i^j) \cdot (a_i^k - k, 1).$$

То есть для перехода с j -го слоя к k -му, нужно найти максимальное скалярное произведение из предыдущих векторов вида $(a_i^j, W_i^j + j a_i^j)$ с вектором $(a_i^k - k, 1)$. Для этого можно поддерживать выпуклую оболочку в таких точках. В третьей подзадаче достаточно делать это на стеке и убирать последние точки, так как a_i^j монотонно не возрастают.

Для общего решения нужно написать дерево поиска, чтобы находить скалярное произведение, и спускаться по дереву. То есть отдельно поддерживать верхнюю и нижнюю оболочку, для самой левой и правой точки искать скалярное произведение, а затем запускать бинарный поиск и находить максимальное значение. Для обновления оболочки можно использовать динамическую выпуклую оболочку — проверять выше или ниже оболочки находится точка, которую мы хотим вставить, а затем вставить ее в нужное место, удаляя некоторые точки справа и слева от него.

Всё это писать может быть довольно трудоемко, но можно привести $y = a_i^j x + (W_i^j + j a_i^j)$ к линейным функциям и использовать дерево Li Chao для нахождения максимального произведения.

ИТОГИ / FINAL STANDINGS

№ #	Имя, фамилия Name	Страна Country	Команда Team	Задачи / Tasks														ИТОГО TOTAL	НАГРАДА AWARD	
				1-й день / 1st							2-й день / 2nd									
				A	B	C	D	E	F	G	H	I	J							
1	Kryukov Maksim	Russian Federation	Comprehensive School at the Center for Pedagogical Excellence	100	100	100	100	28	100	100	100	100	100	100	100	100	100	100	928	ABS, 1DG, HML2
2	Siew Jing Hong	Republic of Singapore	NUS High School of Mathematics and Science	100	100	100	100	17	100	100	100	100	100	100	100	100	100	100	917	1DG, HML2
3	Bessolitsyn Maksim	Russian Federation	Specialized Educational Scientific Center Of Ural Federal University named after the 1st President of Russia B.N. Yeltsin	100	100	100	45	100	100	45	100	100	45	81	100	100	100	871	1DG	
4	Roberto Eugenio Sugiarto	Republic of Indonesia	Simetri	100	100	100	100	28	100	100	100	100	100	100	100	100	100	11	839	1DG
5	Musalimov Olzhass	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	100	100	100	100	28	100	100	100	100	100	100	100	100	100	11	839	1DG
6	Gunadi Gani	Republic of Indonesia	Wardaya College	50	100	100	100	28	100	100	100	100	100	100	100	100	100	33	811	1DG
7	Golikov Egor	Russian Federation	Specialized Educational Scientific Center Of Ural Federal University named after the 1st President of Russia B.N. Yeltsin	100	100	100	45	17	100	100	100	100	85	11				758	2DG	
8	Khaev Bulat	Russian Federation	Advanced Educational Center IT-Lyceum of Kazan Federal University	0	100	100	30	28	100	100	100	100	100	100	100	100	100	758	2DG, HML2	
9	Zhilina Darya	Russian Federation	Lyceum Boarding School No. 61 for Gifted Children	100	100	100	45	28	100	100	45	85	11				714	2DG, HMLA		
10	Nuriyev Amir	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan		100	100	45	28	100	100	100	53	100	100	100	100	100	700	2DG	
11	Tairmas Mansur	Republic of Kazakhstan		100	100	45	28	100	100	100	100	100	100	100	100	100	11	684	2DG	
12	Shumilov Arsen	Russian Federation	I National team of Republic of Sakha (Yakutia)	100	100	97	45	28	100	100	45	50	11				676	2DG		
13	Nathan Allan	Republic of Indonesia	Simetri	100	35	100	45	28	52	100	100	100	100	100	100	100	11	671	2DG	

№ #	Имя, фамилия Name	Страна Country	Команда Team	Задачи / Tasks												ИТОГО TOTAL	НАГРАДА AWARD
				1-й день / 1st						2-й день / 2nd							
				A	B	C	D	E	F	G	H	I	J				
14	Yerlanuly Nursultan	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	100	100	100	100	28	37	100	100	85	11	661	2DG		
15	Leskova Anastasiya	Russian Federation	Specialized Educational Scientific Center Of Ural Federal University named after the 1st President of Russia B.N. Yeltsin	100	100	100	45	28	100	100	15	50	0	638	3DG, HMLA		
16	Nuraly Serikbay	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	100	100	45	28	100	100	45	100	11	629	3DG		
17	Turaev Pavel	Russian Federation	Lyceum Boarding School No. 61 for Gifted Children	100	29	100	45		37	100	100	100	11	622	3DG		
18	Suleymanov Karim	Russian Federation	Advanced Educational Center IT-Lyceum of Kazan Federal University	100	29	100	45	28	100	100	45	50		597	3DG		
19	Sitdikov Damir	Russian Federation		0	29	100	45	41	100	100	100	65	11	591	3DG		
20	Kaztayev Alisher	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	29	36	45	28	100	100	100	50	53	541	3DG		
21	Sveshnikov Boris	Russian Federation	Specialized Educational Scientific Center Of Ural Federal University named after the 1st President of Russia B. N. Yeltsin	0	29	100	45	28	100	100	45	50	11	508	3DG		
22	Kotlychkov Egor	Russian Federation		100	29	0	45	28	52	100	45	50		449	HMLP		
23	Ismagzi Isatay	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	100	0	45	28	37	100	45	76	11	442	HMLP		
24	Nechaev Gleb	Russian Federation	Specialized Educational Scientific Center Of Ural Federal University named after the 1st President of Russia B.N. Yeltsin	10	100		45		100	100	30	50		435	HMLP		
25	Kamiyev Zhanserik	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	29	100	45	28	20	100	45	50	11	428	HMLP		
26	Darius Chitu	Romania	Andrei Şaguna National College	100	29	100	0	28	100	20	15	13	11	416	HMLP		
27	Kulichkin Timur	Russian Federation	I National team of Republic of Sakha (Yakutia)	100	35		45	17	100	100	15			412	HMLP		

№ #	Имя, фамилия Name	Страна Country	Команда Team	Задачи / Tasks														ИТОГО TOTAL	НАГРАДА AWARD		
				1-й День / 1st							2-й День / 2nd										
				A	B	C	D	E	F	G	H	I	J								
28	Vinogradov Semen	Russian Federation	Specialized Educational Scientific Center Of Ural Federal University named after the 1st President of Russia B. N. Yeltsin	100	29	36	30	17	52	100	15	30							409	HMLP	
29	Ivchenko Andrei	Russian Federation		100	29		45	17	37	20	100	50							398		
30	Mazhitov Dzhafar	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	35	29	36	45	17	100	55	15							332			
31	Semenov Artem	Russian Federation	I National team of Republic of Sakha (Yakutia)	35			45		100	100	37							317			
32	Akhmetov Alisher	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	35	100	15	28	37	20	15	50							300		
33	Grigorev Mikhail	Russian Federation	I National team of Republic of Sakha (Yakutia)	100	29		15		37	100	15							296			
34	Mindaev Damir	Russian Federation		75	29	0		11	100	0	15	50							280		
35	Shilov Yaroslav	Russian Federation	Specialized Educational Scientific Center North-Caucasus Federal University	100	0	100			37	10	15							262			
36	Ferrel Himawan Handoyo	Republic of Indonesia	Wardaya College	0	29		45	17	100	10	15	34							250		
37	Sultanbek Zhumagali	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	7		15		100	100	15	13							250		
38	Toktasynov Madi	Republic of Kazakhstan		0	7		45	0	100	55	0	23	0							230	
39	Likhachev Konstantin	Russian Federation	Specialized Educational Scientific Center Of Ural Federal University named after the 1st President of Russia B. N. Yeltsin	10	7	9	45		100	20	15	19	0							225	
40	Shyntai Imanali	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	7		45	28	100	10	15							205			
41	Yumshanov Erchim	Russian Federation	II National team of Republic of Sakha (Yakutia)		29				100	20	15	30							194		

№ #	Имя, фамилия Name	Страна Country	Команда Team	Задачи / Tasks												ИТОГО TOTAL	НАГРАДА AWARD								
				1-й День / 1st						2-й День / 2nd															
				A	B	C	D	E	F	G	H	I	J												
42	Sipattalov Islam	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	29		0	28	37	10	15	30											149		
43	Justin Julius Tan	Republic of Indonesia	Wardaya College	0	29		30	17	37	0	15	16	0											144	
44	Dyachkovskii Damir	Russian Federation	I National team of Republic of Sakha (Yakutia)		7	24	45					15	44											135	
45	Nikolaev Vladimir	Russian Federation		7			45	17	37	0	15													121	
46	Tolstykh Georgii	Russian Federation	Specialized Educational Scientific Center Of Ural Federal University named after the 1st President of Russia B.N. Yeltsin		35		45					20	15											115	
47	Antonov Aleksandr	Russian Federation	II National team of Republic of Sakha (Yakutia)	0	29			17	4	55														105	HMLY
48	Sandhya Mahendra Dhaneswara	Republic of Indonesia		Wardaya College		29		30			37	0												96	
49	Gavrilev Duolan	Russian Federation	I National team of Republic of Sakha (Yakutia)		29		0	17	16	10	15													87	
50	Oteu Beibarys	Republic of Kazakhstan	The Republican Scientific and Practical Center "Daryn" of the Ministry of Education of the Republic of Kazakhstan	0	7		0	37	0	15	9													68	
51	Andi Saliman Najafy	Republic of Indonesia	Wardaya College	0	7	0	15	17	10	0	0	13	0											62	
52	Abraham Imanuel Kaloh	Republic of Indonesia																						0	

НАГРАДЫ / AWARDS:

ABS – абсолютный победитель / absolute winner,

1DG – диплом 1 степени / 1st degree diploma,

2DG – диплом 2 степени / 2nd degree diploma,

3DG – диплом 3 степени / 3rd degree diploma,

HML2 – почетная грамота за полное решение задачи 2 дня / Honourable mention letter "For the perfect result of Day 2",

HMLP – почетная грамота за многообещающие результаты / Honourable mention letter "For the promising results",

HMLA – почетная грамота "По стопам Ады Лавлейс" / Honourable mention letter "In the footsteps of Ada Lovelace",

HMLY – почетная грамота "Самому юному участнику" / Honourable mention letter "For the youngest participant",

Contents | Содержание

Problem statements	1
A. Golfilliard	1
B. The librarian-robot	2
C. I'm the Map	3
D. A game for the gamer-student	5
E. Replacing characters	7
F. The City View	9
G. Magnets on the board	10
H. Schedule	12
I. Ring Road	13
J. The Thought	15
Solutions	18
A. Golfilliard	18
B. The librarian-robot	18
C. I'm the Map	19
D. A game for the gamer-student	20
E. Replacing characters	20
F. The City View	21
G. Magnets on the board	22
H. Schedule	22
I. Ring Road	23
J. The Thought	23
Задачи	25
A. Гольфильярд	25
B. Робот-библиотекарь	26
C. Я – Карта	28
D. Игра для студента-геймера	30

E.	Замена символов	31
F.	Вид на город с разных сторон	33
G.	Магнитики на доске	35
H.	Расписание	36
I.	Кольцевая дорога	38
J.	Мысль	40
Решения		43
A.	Гольфильярд	43
B.	Робот-библиотекарь	43
C.	Я – Карта	44
D.	Игра для студента-геймера	45
E.	Замена символов	46
F.	Вид на город с разных сторон	47
G.	Магнитики на доске	47
H.	Расписание	48
I.	Кольцевая дорога	48
J.	Мысль	49

Final Standings | Итоги олимпиады