西安電子科技大學 XIDIAN UNIVERSITY 程序设计竞赛实训基地 Programming Contest Training Base

2020-2021 "Orz Panda" Cup Programming Contest

Sponsored by

School of Aerospace Science and Technology

Xidian University

China, Xi'an

November 22, 2020

This problem should contain 10 problems on 20 pages.
Please inform a runner immediately if something is missing from your problem set.

Problem A. Accordion Artist And Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.3 seconds
Memory limit:	256 megabytes

Chuan *the Country Builder* is a famous accordion artist. Now he wants to put a "Make Accordion Great Again" advertisment, in order to get some advantage over his enemy Bai *the Sleeper*.

There are n buildings on the 114514th Avenue, the Old York city. The premium alongside the 114514th Avenue is very high so there is no margin between two adjacent buildings. The front view of the i-th building can be approximated with a rectangle with height h_i and width w_i.

Chuan wants his advertisment to be a *great* one. More specifically, it must be a rectangle with the height perpendicular to the height of the front view of each building, and the entire advertisment must fit in the union of the front views of the buildings.

An Orz Panda company has made a contract to build the advertisment for Chuan. The company wants the area of the advertisment to be as large as possible, so they can get more money from Chuan. Please calculate the maximum possible area of all *great* advertisments.

Input

There is only one test case in the input file.

The first line contains an integer n, the number of the buildings.

The i-th of the following n lines contains two integers h_i, w_i, the height and weight of the i-th building.

It's guaranteed that $1 \le n \le 10^5$, and $1 \le h_i, w_i \le 10^5$.

Output

Output one line containing the maximum area.

Examples

standard input	standard output
5 2 1 4 1 3 1 4 1 1 1	9
5 2 1 4 1 3 1 4 1 2 1	10

Note

The *great* advertisment with maximum area is illustrated in the following figure. The grey area is the front view of the buildings. There is some margin deliberately added around the advertisment, to make the figure clear.


I will
MAKE ACCORDION
GREAT AGAIN!

Problem B. Bracelets of Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Master jl0x62 loves bracelets very much. One day he travals to the land of Orz Pandas and goes to visit a souvenir shop. The Orz Pandas are selling a special kind of bracelets, named *the Orz Panda Bracelet* (OPB) here. An OPB is a bracelet (cylindrical face) with length n, height 2, and composed by some 1×2 rectangular blocks. Certainly there would be n blocks, for an OPB with length n.

Master jl0x62 will pay the Orz Pandas n dollars for each OPB with length n. But he'll only pay for unique OPBs. Two OPBs are considered same if one coincides with another after a rotation about the axis passing through the center of the OPB and parallel in the direction of the OPB's height.

Due to a technical limitation, the Orz Pandas can only make OPBs with its length n not greater than m. Please tell the Orz Pandas how many dollars they can get by selling the OPBs to Master jl0x62.

The Orz Pandas hate multi-precision numbers. So you should output the answer modulo an integer p.

Input

There is only one test case in each input file.

The test case is a single line, containing two integers m and p seperated by a whitespace.

It's guaranteed that $1 \le m \le 10^9$, $1 \le p \le 10^9 + 9$.

It's **not** guaranteed that p is a prime.

Output

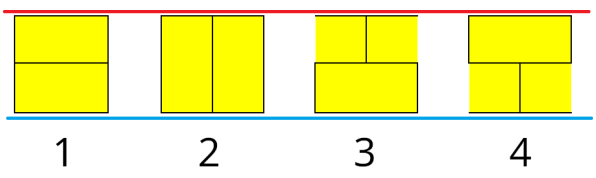
For each test case, output one line, containing the maximum amount of dollars the Orz Panda can get by selling the OPBs to jl0x62, modulo p.

Examples

standard input	standard output
1 114514	1
2 1919810	7
3 1000000007	13
4 998244353	29

Note

For example, if $m = 2$, the Orz Pandas can make 1 unique OPB with length 1, and 3 unique OPBs with length 2. So they can get $1 \times 1 + 3 \times 2 = 7$ dollars.



The figure above shows the OPBs with length 2. The subfigures labeled 1, 2, and 3 represent three unique OPBs. The subfigure labeled 4 represents the same OPB as subfigure 3, but with a different cutting edge.

This page is intentionally left blank

Problem C. Closestools of Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	256 megabytes

There is a restroom for male Orz Pandas with n closestools, labeled $1, 2, \cdots, n$. The distance between the closestool i and the closestool j is $|i - j|$.

The Orz Pandas are somewhat self-closed. When an Orz Panda enters the restroom, he will choose a closestool x, to make the distance between x and any other closestools already occupied by another Orz Panda as large as possible.

If there are mulitple closestools satisifying this condition, the Orz Panda will choose the one with the smallest label.

Please write a program to simulate the operation of this restroom.

Input

There are multiple test cases. Please process until EOF.

For each test case, the first line contains two integers n and m, where n is the number of closestools, and m is the number of operations. Each of the next m lines contains an operation. An operation may be one of the following:

- `1`: an Orz Panda enters the restroom
- `2 x`: the Orz Panda entered the restroom at the x-th operation leaves the restroom

It's guaranteed that $1 \le n \le 10^9$, $1 \le m \le 10^5$, and the sum of m in all test cases will not exceed 10^6. For each type `2` operation, it's guaranteed that x is unique and the x-th operation is always a previous type `1` operation. It's guaranteed that the number of Orz Pandas in the restroom will not exceed n at any time.

Output

For each type `1` operation, output one line, containing the label of the closestool the entering Orz Panda will choose.

Example

standard input	standard output
7 10 1 1 1 1 1 2 3 1 2 4 2 5 1	1 7 4 2 3 5 3

This page is intentionally left blank

Problem D. Data Structure Master and Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Master wang9897 is the First Data Structure Master in Xidian University. Today he is teaching some Orz Pandas to learn Link-Cut Tree (LCT). As a practice, wang9897 has given an interesting problem to the Orz Pandas.

Note that you don't need to know LCT to solve this problem.

Given a rooted tree with n nodes whose root is numbered 1. In LCT each node u can be assigned one *preferred child* v. Certainly v must be a child of u. Initially there is no preferred child assigned to each node.

Then m "LCT access" operations are performed as follow. For each operation, a node v is chosen from all nodes uniformly. The path from the root to v, $(v_1, v_2, \cdots, v_k)$ is then found, where $v_1 = 1$, and $v_k = v$. For $i = 1, 2, \cdots, k-1$, if v_i is not assigned a preferred child, or it is assigned to a preferred child which is not v_{i+1}, the preferred child of v_i is reassigned to v_{i+1}.

Let $f(m)$ be the time of reassignments of preferred child in total. Master wang9897 want the Orz Pandas to calculate

$$\lim_{m \to \infty} \frac{f(m)}{m}$$

Please solve the problem so wang9897 can compare the Orz Pandas' answer to yours.

Input

There is only one test case in the input file.

The first line contains a integer n, the size of the tree. The second line contains $n-1$ integers $p_2, p_3, \cdots, p_{n-1}$. p_i is the parent of the node i.

It's guaranteed that $1 \le p_i \le n \le 10^5$.

Output

Output one line, containing an integer, the answer modulo 998244353.

Example

standard input	standard output
5 1 1 2 2	499122177

Note

For the sample, the answer is $1/2$, so $1 \cdot 2^{-1}$ should be outputed.

This page is intentionally left blank

Problem E. Encryption of Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

An Orz Panda agent has some data which should be encrypted. He has store the data into a list `a` with length `n`. To encrypt it, he'll run the following loop (wrote in Python) for k times:

```
for i in range(1, n):
  a[i] ^= a[i-1]
```

Please output the content of the list `a` after the encryption.

Input

There is only one test case in the input file.

The first line contains two integers n and k, seperated by a whitespace.

The second line contains n integers $a_0, a_1, \cdots, a_{n-1}$, the elements in the list `a`. Each adjacent pair of integers are seperated by a whitespace.

It's guaranteed that $1 \leq n \leq 10^5$, $0 \leq a_i < 2^{17}$, and $1 \leq k \leq 10^{18}$.

Output

Output one line, containing n integers in the encrypted list `a`. Each adjacent pair of integers should be seperated by a whitespace. There should be no trailing whitespaces.

Example

standard input	standard output
6 5 1 0 5 1 5 5	1 1 4 5 1 4

Note

For the example:

- The list should become `[1,1,4,5,0,5]` after the first iteration
- The list should become `[1,0,4,1,1,4]` after the second iteration
- The list should become `[1,1,5,4,5,1]` after the third iteration
- The list should become `[1,0,5,1,4,5]` after the fourth iteration
- The list should become `[1,1,4,5,1,4]` after the fifth iteration, and you should output it

This page is intentionally left blank

Problem F. Flow of Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are n villages in a county of the Orz Pandas. The i-th village needs w_i tons of water per day. The Orz Pandas have built k water supply facilities. The i-th water supply facility is located at the s_i-th village. The water supply facilities are very powerful, so we could consider each of them able to supply infinite amount of water per day.

The Orz Pandas have built water supply pipe networks to satisify the water usage of the villages. There are m water supply pipes already built. The i-th pipe is between the village u_i and the village v_i. The pipes are bidirectional: if there is a pipe between u_i and v_i, it's allowed to transfer water from u_i to v_i, or from v_i to u_i, through this pipe. Pipe i has a parameter c_i. If f_i tons of water are transfered through the i-th pipe per day, the Orz Pandas will have to pay $f_i^2 \times c_i$ dollars per day to maintain the pipe.

Please tell the Orz Pandas the minimum cost they must pay per day, to satisify the water usage of all the villages.

Input

There is only one test case in each input file.

- The first line contains three integers n, m, and k.
- The second line contains n integers $w_1, w_2, \cdots, w_n$.
- The third line contains k integers $s_1, s_2, \cdots, s_k$.
- The i-th of the following m lines contains three integers u_i, v_i, and c_i.

It's guaranteed that $1 \le n \le 50$, $0 \le m \le 200$, $1 \le k, s_i, u_i, v_i \le n$, and $0 \le w_i, c_i \le 1000$.

There **may** be multiple pipes connecting the same pair of villages. There **may** be some pipe both endings of which are the same village. There **may** be multiple water supply facilities in the same village.

Output

If the water usage of all the villages can be satisified, output one line containing a decimal, which is the minimum cost. Otherwise, output one line containing -1. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-9}.

Examples

standard input	standard output
7 5 2 0 0 0 0 1 1 0 1 3 1 2 1 3 4 2 4 2 1 2 5 2 4 6 1	5.75
7 5 2 0 0 0 0 1 1 1 1 3 1 2 1 3 4 2 4 2 1 2 5 2 4 6 1	-1

Note

For the first sample, the optimal solution is:

- Transfer 1.25 tons of water from village 1 to village 2 per day, costing $1.25^2 \times 1 = 1.5625$ dollars per day.
- Transfer 0.75 tons of water from village 3 to village 4 per day, costing $0.75^2 \times 2 = 1.125$ dollars per day.
- Transfer 0.25 tons of water from village 2 to village 4 per day, costing $0.25^2 \times 1 = 0.0625$ dollars per day.
- Transfer 1 ton of water from village 2 to village 5 per day, costing $1^2 \times 2 = 2$ dollars per day.
- Transfer 1 ton of water from village 4 to village 6 per day, costing $1^2 \times 1 = 1$ dollars per day.

For the second sample, it's impossible to satisify village 7 since there is no way to connect it to a water supply facility.

Problem G. Gery's Problem and Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Gery got "`Compilation Error`" because he forgot to include the C++ standard header `cstring`. So he lost all the score of three problems. One of the problems is:

There is a tree with n verticies. m queries would be performed on it. Each query contains two parameters u and v. You should calculate

$$\sum_{r=1}^{n} d_r(u,v)$$

$d_r(u,v)$ is the *beautifulness* of the road from vertex u to vertex v in the tree, with vertex r as the root.

With vertex r as the root, the *beautifulness* of the road from u to v is defined

$$d_r(u,v) = dis(u, lca_r(u,v)) \times dis(v, lca_r(u,v))$$

$dis(u,v)$ is the direct distance between u and v, and $lca_r(u,v)$ is the least common ancestor of u and v, with r as the root.

Originally the constraint of this problem was $1 \le n, m \le 1000$. But Gery was in a bad mood so he fortified the problem. The constraint is now $1 \le n, m \le 10^5$.

Help the Orz Pandas to solve this problem so they would be able to access `https://gery.top`.

Input

There is only one test case in the input file.

The first line contains two integers n and m . The i-th of the following $n-1$ lines contains two integers x_i and y_i, the vertices connected by the i-th edge. Each of the following m lines contains a pair of integers, u and v, the pair of vertices being queried.

It's guaranteed that $1 \le n, m \le 10^5$, and $1 \le x_i, y_i, u, v \le n$.

Output

For each query, output one line containing the answer of the query, modulo 998244353.

Example

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

Note

For the second query, we have $d_1(2,5) = dis(2,1) \times dis(5,1) = 2$, $d_2(2,5) = dis(2,2) \times dis(5,2) = 0$, $d_3(2,5) = dis(2,3) \times dis(5,3) = 2$, $d_4(2,5) = dis(2,3) \times dis(5,3) = 2$, and $d_5(2,5) = dis(2,5) \times dis(5,5) = 0$. Their sum is 6.

This page is intentionally left blank

Problem H. Hamming Code and Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Chuan *the Country Builder* is a famous accordion artist. He planned to hold a concert in the Orz Panda's Land. Unfortunately he is just cured from a COVID-19 infection so he would be rejected by the Border Control Administration of the Orz Panda.

As an alternative, Chuan recorded his accordion songs as a binary data flow and sent the binary data to the Orz Pandas. However Chuan's enemy, Bai *the Sleeper* can use a magic to modify the binary data being transfered. To prevent Bai's disruption, Chuan used Hamming code to encode the binary data.

A data block using Hamming code d has 2^k bits, numbered $0, 1, \cdots, 2^k - 1$. Let $f(x)$ be the number of ones in the binary form of the integer x, for example $f(10) = 2$ since $10 = 1010_2$. To use Hamming code, Chuan can only put data into those positions x with $f(x) > 1$. If we name the bits at those postions "data bits", there are only $2^k - k - 1$ data bits.

For example, if $k = 4$, the size of d will be $2^k = 16$, and there are $2^4 - 4 - 1 = 11$ data bits. If the data is 10110111011, the original data block is like:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$d(i)$	?	?	?	1	?	0	1	1	?	0	1	1	1	0	1	1

Then the remaining $k + 1$ non-data bits (marked as ? above) should be filled. Firstly, for each position $j \in [0, k)$, the 2^j-th bit is caluculated as the xor-sum:

$$d(2^j) = \bigoplus_{(i \cap 2^j) \neq 0, i \neq 2^j} d(i)$$

$\cap$ is the binary-and operator. For example, in the data block above we have:

$$d(2^0) = d(3) \oplus d(5) \oplus d(7) \oplus d(9) \oplus d(11) \oplus d(13) \oplus d(15) = 0$$
$$d(2^1) = d(3) \oplus d(6) \oplus d(7) \oplus d(10) \oplus d(11) \oplus d(14) \oplus d(15) = 1$$
$$d(2^2) = d(5) \oplus d(6) \oplus d(7) \oplus d(12) \oplus d(13) \oplus d(14) \oplus d(15) = 1$$
$$d(2^3) = d(9) \oplus d(10) \oplus d(11) \oplus d(12) \oplus d(13) \oplus d(14) \oplus d(15) = 1$$

Secondly, the 0-th bit is calculated as the xor-sum:

$$d(0) = \bigoplus_{i>0} d(i)$$

For example, in the data block above we have $d(0) = 1$. At last this data block shall become

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
d(i)	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1

Then Chuan would transfer the encoded data block to the Orz Pandas. Bai will attempt to disrupt the data transfer, but he is too sleepy so he can only change at most 2 bits in a block. It can be proved that the Orz Pandas can always detect if the block is disrupted by Bai. And it can be proved that if Bai only changed 1 bit in a block, the Orz Pandas can find exactly the bit which is changed. You should write a program to implement it for the Orz Pandas.

Input

There are multiple test cases, please process until EOF.

For each test case, the first line contains an integer k. The second line contains 2^k binary bits $d(0), d(1), \cdots, d(2^k - 1)$.

It's guaranteed that $3 \leq k \leq 16$, $d(i) \in \{0, 1\}$, and at most 2 bits are changed by Bai. The total number of binary bits in the input file will not exceed 1048576.

You should take care that Bai **may** change non-data bits.

Output

For each test case output one line. If the block is not disrupted, output "`good`". If the block is disrupted and 2 bits are changed, output "`broken`". If the block is disrupted and only 1 bit is changed, output "`d(x) is changed`", where "`x`" should be replaced by the position of the only changed bit.

Example

standard input	standard output
4 1011101110111011 4 1011101110111010 4 1011101110111110	good d(15) is changed broken

Problem I. Irregular Shape of Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Some Orz Pandas are watching Master Ma's video and laughing. Master Ma is a famous combat trainer. In the video he is presenting his signature move, *the Five Continous Whips of Lightning*. He has sketched the move on a blackboard. The shape of the sketch is very irregular but can be approximated as a polygon.

Master Ma believes the power of *the Five Continous Whips of Lighting* is proportional to the area of this polygon. While the Orz Pandas are laughing, they also want to know the value of this area.

Input

There are multiple test cases. Please process until EOF.

For each test case, the first line contains one integer n, the number of vertices of the polygon. The i-th of the next n lines contains two integers x_i and y_i, the Cartesian coordinates of the i-th vertex.

The i-th edge of the polygon is the segment connecting the i-th vertex and the $(i+1)$-th vertex. The n-th edge is the segment connecting the n-th and the first vertex.

It's guaranteed that $3 \leq n \leq 10^3$, $-10^9 \leq x_i, y_i \leq 10^9$, and the sum of n of all test cases in the input file will not exceed 10^4.

Since Master Ma's move is too irregular, the polygon may be **non**-convex. But it's guaranteed that the polygon is simple: each line segment endpoint is shared by exactly two segments, and the segments do not otherwise intersect. It's also guaranteed that all vertices are unique.

Output

For each test case output one line containing the area of the polygon. The area should be rounded up to exactly two decimal places after the point.

This problem is **not** special judged.

Examples

standard input	standard output
3 1 1 4 5 1 4	4.50
6 0 0 1000000000 0 1000000000 999999999 999999999 999999999 999999999 1000000000 0 1000000000	999999999999999999.00
12 -1000000000 -1000000000 1000000000 -1000000000 1000000000 1000000000 -1000000000 1000000000 -1000000000 -999999998 999999998 -999999998 999999998 -999999997 -999999999 -999999997 -999999999 999999999 999999999 999999999 999999999 -999999999 -1000000000 -999999999	9999999992.00
12 -1000000000 1000000000 1000000000 1000000000 1000000000 -1000000000 -1000000000 -1000000000 -1000000000 999999998 999999998 999999998 999999998 999999997 -999999999 999999997 -999999999 -999999999 999999999 -999999999 999999999 999999999 -1000000000 999999999	9999999992.00

Note

To reduce the number of "`Wrong Answer`", the Orz Pandas added several additional examples. Use them wisely.

Problem J. Junction of Orz Pandas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

The Orz Pandas are building a electric power grid. There are n vertical power lines and m horizontal power lines. To make the power grid stable, the Orz Pandas should build one junction between each vertical power line and each horizontal power line. There should be $n \times m$ junctions.

Each junction has a design parameter. Let's use p_{ij} to denote the design parameter of the junction between the i-th vertical power line and the j-th horizontal power line. To ensure the safety of the power grid, it must satisfies the following conditions:

- p_{ij} is a positive integer
- $a_i = \max\limits_{1 \le j \le m} \{p_{ij}\}$, for each i
- $b_j = \max\limits_{1 \le i \le n} \{p_{ij}\}$, for each j

Now the Orz Panda engineers have already calculated a_i and b_j for each power line. It's very easy to output a possible solution of p_{ij} satisifying those conditions, or report they can't be satisified. To make the problem more challenging, the Orz Pandas want you to output the number of different solutions satisifying those conditions.

Two solutions p and q are considered different if and only if there exists a pair (i, j), such that $p_{ij} \neq q_{ij}$. Since the answer may be very large, you should output it modulo 998244353.

Input

There is only one test case in the input file.

The first line contains two integers n and m.

The second line contains n integers $a_1, a_2, \cdots, a_n$.

The third line contains m integers $b_1, b_2, \cdots, b_n$.

It's guaranteed that $1 \le n, m \le 10^5$, and $1 \le a_i, b_i \le 10^9$.

Output

Output one line containing the number of different answers, modulo 998244353.

Example

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

Note

For the sample, the possible solutions are:

1	1	1
1	2	1
1	1	3

1	1	1
1	2	2
1	1	3

1	1	1
1	2	1
1	2	3

1	1	1
1	2	2
1	2	3

1	1	1
1	1	2
1	2	3

This page is intentionally left blank