Problem A. Beta Go

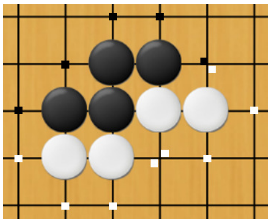
Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Go is a traditional board game for two players invented in China more than 2500 years ago. In this game, the aim is to surround more territory than the opponent.

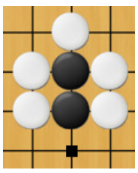
The two players, Black and White, take turns placing stones of their color on the intersections of the board, one stone at a time. The board size is a $n \times n$ grid. In the beginning, the board is empty. Black plays first. The players may choose any unoccupied intersection to play on, except for those forbidden by the suicide rules (see below). Once played, a stone can never be moved. A player may also pass, declining to place a stone. When both players pass consecutively, the game ends and is then scored.

Liberties: Vertically and horizontally adjacent stones of the same color form a group, a discrete unit that cannot then be divided. Notice that only stones connected to one another by the lines on the board create a group, but stones that are diagonally adjacent are not connected. A vacant point adjacent to a stone, along one of the grid lines of the board, is called a liberty for that stone. Stones in a group share their liberties. A group of stones must have at least one liberty to remain on the board. When a group is surrounded by opposing stones so that it has no liberties, it is captured and removed from the board.

There are one black group and two white groups in the picture below, with their liberties marked with dots. Liberties are shared among all stones of a group and can be counted. Here the black group has 5 liberties, while the two white groups have 4 liberties each.

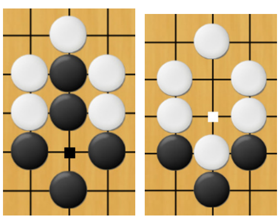


Capture: In the picture below, the black group has only one liberty. If White plays at the black dot, the two black stones will be removed from the board, which is called capture.

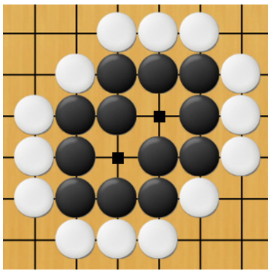


Suicide Rule: A player may not place a stone such that it or its group immediately has no liberties, unless doing so immediately captures an enemy group of its final liberty. In the latter case, the enemy group is captured, leaving the new stone with at least one liberty.

In the picture below, if White plays a stone at the black dot, which has no liberty either, the two black stones will be immediately removed from the board, and then the stone White just placed has one liberty.



However, in the picture below, White can not play a stone at either of the two black dots. Because the stone White places has no liberty, and the black group has one liberty left.



Beta Go is a computer program that plays the board game Go very well, and it has defeated many professional human Go players. Moca loves playing the board game, and Go is her favorite. One day, Moca is playing Go with Beta Go online. Of course, she never wins Beta Go, which makes her very angry. As a result, after Beta Go placing several stones on the board, Moca hacks into the game system to stop Beta Go from playing and removes the stones she placed on the board. Now, the only stones left on the board are the ones Beta Go played, and Moca continues to place her stones on the board, while Beta Go can only choose to pass its turn.

Your task is to help Moca capture the maximum number of stones that Beta Go played. Notice that you can not break the rules mentioned above, and you can not move the stones Beta Go played except capturing.

Input

The first line contains one integer n ($1 \le n \le 1000$) – the size of the board.

The next n lines describe the game board. Each of the following n lines contains n characters '`.`' or '`O`'. If a character equals '`.`', the corresponding intersection is empty. If it equals '`O`', the corresponding cell is a stone Beta Go played.

It is guaranteed that there exists **at least one empty cell**.

Output

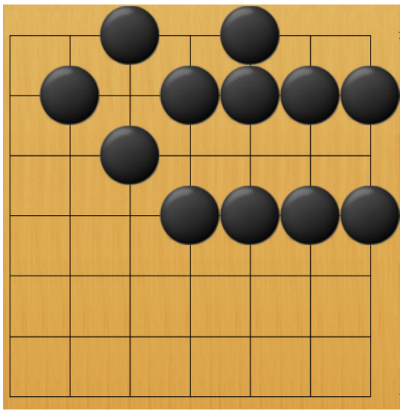
Print one integer – the maximum number of stones captured.

Examples

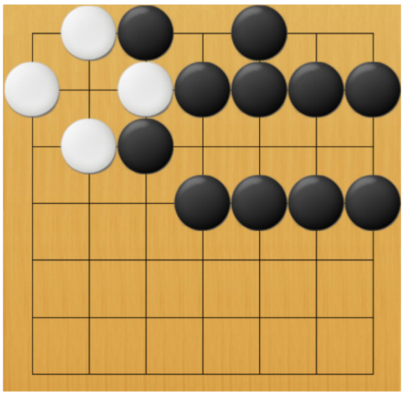
standard input	standard output
7 ..O.O.. .O.OOOO ..O.... ...OOOO	12
5 .O.O. .OOOO ....O	1

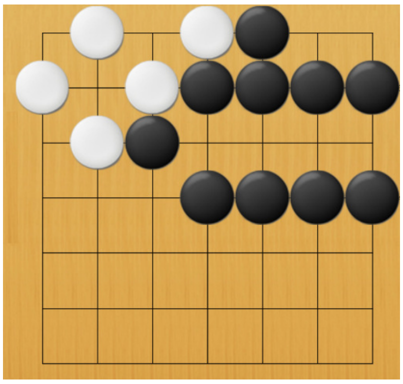
Note

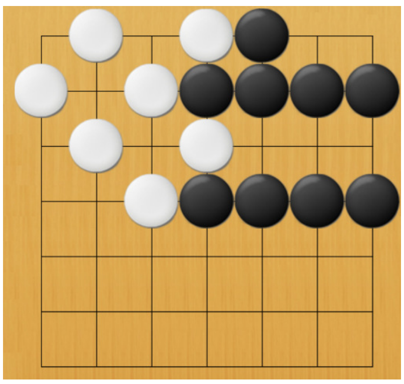
In the first example, assume that Beta Go plays black stones. The initial game board is as below.



In the beginning, you can not capture the upper right black group, but you can capture these three black stones in the next three pictures in order.







Now, you can easily capture the last two black groups. Finally, you can capture all 12 stones that Beta Go played.

Problem B. Build Roads

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

In the cat country, there are n cities. The cat country king wants to build $n-1$ roads to connect all the cities. The i-th city has a construction company with an experience value of a_i. To build a road between the i-th city and the j-th city, the two cities' construction companies need to cooperate with each other. However, in the process of building a road, the two construction companies may have conflicts due to poor communication, and it will result in a waste of building materials. Formally, building a road between the i-th city and the j-th city will waste $\gcd(a_i, a_j)$ building materials.

Can you help the cat country king choose to build $n-1$ roads that connect all the cities and minimize the waste of building materials?

To decrease the input size, the king of the cat country gives you a random integer generator and 3 parameters $L, R, seed$. The following C language code shows how to generate n integers $a_1, a_2, \ldots, a_n$, and $a[i]$ stores the experience value of the construction company of the i-th city. You can use the code directly in your submissions.

```c
#include <stdio.h>

int n, L, R, a[200001];
unsigned long long seed;

unsigned long long xorshift64() {
  unsigned long long x = seed;
  x ^= x << 13;
  x ^= x >> 7;
  x ^= x << 17;
  return seed = x;
}

int gen() {
  return xorshift64() % (R - L + 1) + L;
}

int main() {
  scanf("%d%d%d%llu", &n, &L, &R, &seed);
  for (int i = 1; i <= n; i++) {
    a[i] = gen();
  }
  // ...
}
```

Input

The only line contains four integers $n, L, R, seed$ ($2 \le n \le 2 \times 10^5, 1 \le L \le R \le 2 \times 10^5, 1 \le seed \le 10^{18}$).

Output

Print one integer – the minimum waste of connecting all the cities.

Example

standard input	standard output
5 1 200000 123	4

Problem C. Cat Virus

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

In the cat country, any cat family can be regarded as a **rooted tree**. As we all know, a kind of zombie virus hides in all the cats' bodies. Therefore, a cat family may consist of cats and zombies. When a cat is born, it may become a zombie. If a cat becomes a zombie, all of its offspring will also become zombies.

Now given an integer K, you should construct a cat family with exactly K ways to mark the identities (cat or zombie) of members of this family. Two ways are considered different if and only if there is at least one member that is marked as a cat in one way and marked as a zombie in the other way.

Formally, the vertices in a rooted tree will be marked black or white, and if a vertex is marked black, all the vertices in its subtree should also be marked black. Given a rooted tree, it's easy to calculate the number of possible valid ways to mark the vertices in the tree. Your task is to construct a rooted tree with exactly K ways to mark your tree's vertices. Two ways are considered different if and only if there is at least one vertex such that in one way is marked black and marked white in the other way.

Input

The only line contains an integer K ($2 \le K \le 2 \cdot 10^{18}$) – the number of valid ways to mark the vertices in the tree.

Output

Let's denote the number of vertices of the tree you construct as n, and label the vertices from 1 to n where vertex 1 is the root.

The output should contain n lines. In the first line, print one integer n ($1 \le n \le 10^5$). In the each of the next $n - 1$ lines, print two integers u, v ($1 \le u, v \le n, u \ne v$) describing an edge in your rooted tree.

It is guaranteed that at least one solution exists. If there are several possible valid solutions, you can print any of them.

Examples

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

Problem D. Dyson Box

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

A Dyson Sphere is a hypothetical megastructure that completely encompasses a star and captures a large percentage of its power output. The concept is a thought experiment that attempts to explain how a spacefaring civilization would meet its energy requirements once those requirements exceed what can be generated from the home planet's resources alone. Only a tiny fraction of a star's energy emissions reach the surface of any orbiting planet. Building structures encircling a star would enable a civilization to harvest far more energy.

One day, Moca has another idea for a thought experiment. Assume there is a special box called Dyson Box. The gravitational field in this box is unstable. The direction of the gravity inside the box can not be determined until it is opened.

The inside of the box can be formed as a 2-dimensional grid, while the bottom left corner's coordinate is $(0, 0)$ and the upper right corner's coordinate is $(2 \cdot 10^5, 2 \cdot 10^5)$. There will be n events. In the i-th event, a new cube will appear, whose upper right corner's coordinate is (x_i, y_i) and bottom left corner's coordinate is $(x_i - 1, y_i - 1)$.

There are two directions of gravity in the box, **vertical and horizontal**. Suppose Moca opens the box after the i-th event. In that case, she has $\frac{1}{2}$ probability of seeing the direction of the gravity inside the box is vertical, and the other $\frac{1}{2}$ probability is horizontal. And then, she will measure **the total length of the outline of all the cubes**. If the direction of gravity is horizontal, all the cubes inside will move horizontally to the left under its influence. Similarly, vertical gravity will cause all the cubes to move downward.

Moca hates probability, so that she is asking for your help. If you have known the coordinates of all the cubes in chronological order, can you calculate the total length of these two cases after each event?

Input

The first line contains one integer n $(1 \le n \le 2 \cdot 10^5)$ – the number of cubes.

Each of the following n lines describes a cube with two integers x_i, y_i $(1 \le x_i, y_i \le 2 \cdot 10^5)$.

It is guaranteed that no two cubes have the same coordinates.

Output

For each of the n cubes, print one line containing two integers – two answers when the the direction of gravity is vertical and horizontal.

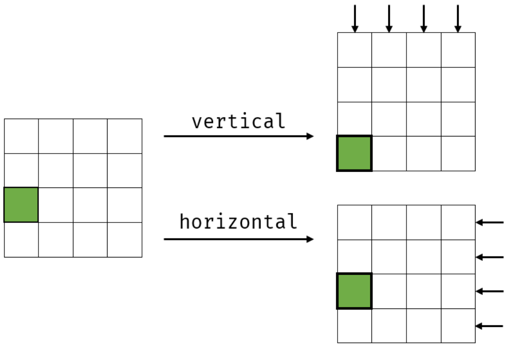
Example

standard input	standard output
4 1 2 3 2 2 1 4 1	4 4 8 6 8 8 10 8

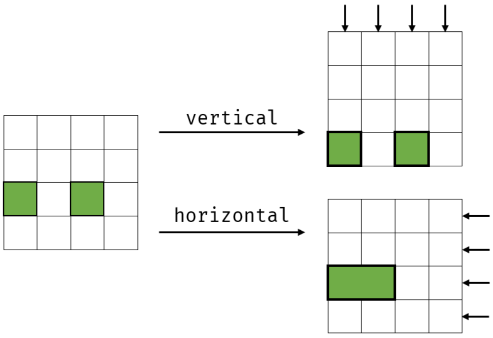
Note

In the only example, the inside of the box is as below, and the bold lines mark the outline of all the cubes.

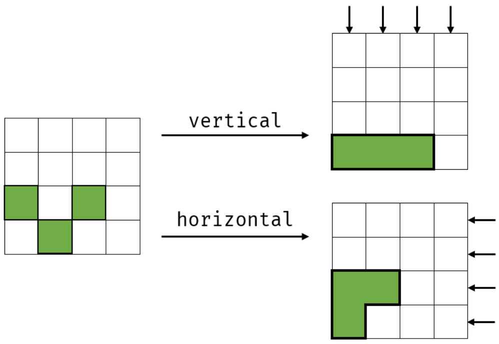
After the 1-st event:



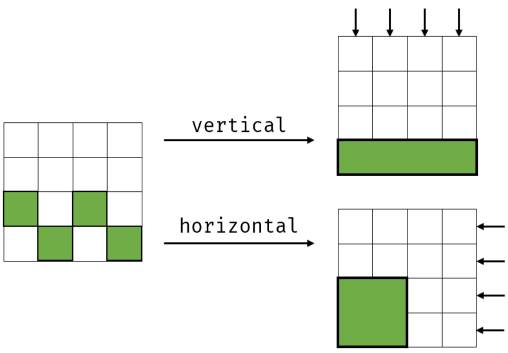
After the 2-nd event:



After the 3-rd event:



After the 4-th event:



Problem E. Evaluate Expression

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Moca is learning how to evaluate a mathematical expression, and she is only able to evaluate a complicated expression step by step. But she is very good at counting, so she is curious about how many different orders evaluate a given expression.

In this problem, you only need to consider some simple mathematical expressions consisting of $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, `+` (addition), `*` (multiplication) and `()`. Multiplication has a higher priority than addition. Parentheses `()` can be used to indicate an alternative order. In each step of the evaluation, Moca can choose an operation (addition or multiplication) and replace it with the corresponding result. Notice that Moca can not break the original priority, which may cause her to evaluate an incorrect result. For example, Moca can not choose to evaluate the middle addition in $1\times1+1\times1$ and the middle multiplication in $(1+1)\times(1+1)$. If many different operations can be evaluated valid, she can choose any one of them.

Moca is very curious. Can you help her count the number of different valid orders to evaluate the given mathematical expression? Since the answer may be very large, you only need to print the answer modulo 998244353.

Input

The first line contains one integer T $(1 \le T \le 100)$ – the number of testcases.

For each testcase, there is only one line contains a valid expression $expr$ $(1 \le |expr| \le 10000)$ consisting of $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, `+`, `*` and `()`. All the numbers that appear are not greater than 9 (only one digit). All the `+` that appear are binary operations (+1 is invalid).

It is guaranteed that the sum of lengths of all expressions does not exceed 20000.

Output

For each testcase, print the number of valid evaluation order modulo 998244353 in one line.

Example

standard input	standard output
3 1+1+1+1 1*2*3+4*5 1+(2*3*4+5+6)*6+7*8*9	6 6 784

Note

In the first example, $1+1+1+1$ has 6 valid evaluation orders:

1. $\underline{1+1}+1+1 \to \underline{2+1}+1 \to \underline{3+1} \to 4$
2. $\underline{1+1}+1+1 \to 2+\underline{1+1} \to \underline{2+2} \to 4$
3. $1+\underline{1+1}+1 \to \underline{1+2}+1 \to \underline{3+1} \to 4$
4. $1+\underline{1+1}+1 \to 1+\underline{2+1} \to \underline{1+3} \to 4$
5. $1+1+\underline{1+1} \to 1+\underline{1+2} \to \underline{1+3} \to 4$
6. $1+1+\underline{1+1} \to \underline{1+1}+2 \to \underline{2+2} \to 4$

In the second example, $1 \times 2 \times 3 + 4 \times 5$ has 6 valid evaluation orders:

1. $\underline{1 \times 2} \times 3 + 4 \times 5 \to \underline{2 \times 3} + 4 \times 5 \to 6 + \underline{4 \times 5} \to \underline{6 + 20} \to 26$
2. $\underline{1 \times 2} \times 3 + 4 \times 5 \to 2 \times 3 + \underline{4 \times 5} \to \underline{2 \times 3} + 20 \to \underline{6 + 20} \to 26$
3. $1 \times \underline{2 \times 3} + 4 \times 5 \to \underline{1 \times 6} + 4 \times 5 \to 6 + \underline{4 \times 5} \to \underline{6 + 20} \to 26$
4. $1 \times \underline{2 \times 3} + 4 \times 5 \to 2 \times 3 + \underline{4 \times 5} \to \underline{2 \times 3} + 20 \to \underline{6 + 20} \to 26$
5. $1 \times 2 \times 3 + \underline{4 \times 5} \to \underline{1 \times 2} \times 3 + 20 \to \underline{2 \times 3} + 20 \to \underline{6 + 20} \to 26$
6. $1 \times 2 \times 3 + \underline{4 \times 5} \to 1 \times \underline{2 \times 3} + 20 \to \underline{1 \times 6} + 20 \to \underline{6 + 20} \to 26$

Problem F. Birthday Cake

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

Moca's birthday is coming up, and her friend Ran is going to the Yamabuki bakery to order a birthday cake for her.

Yamabuki bakery provides n kinds of cake. Since Ran knows that Moca is the Yamabuki Bakery's 1-st fan and can eat a lot of food, she intends to order two cakes and put them together into a big cake. The cake made by Yamabuki bakery can be formed by a string consisting of lowercase Latin letters, which describes the toppings on the cake, such as strawberries and chocolate. Putting two cakes together is the concatenation of two corresponding strings. For example, the result of putting cake `ab` and cake `cabc` together is a big cake `abcabc`.

To make it easier to share the cake, Ran will choose two cakes and put them together into a big cake which can be divided in half into two identical pieces. For example, cake `abcabc` will be divided in half into two cakes `abc`, but cake `abbc` will be divided in half into two different cakes `ab` and `bc`. Notice that Ran can not reverse the cake, which means that cake `abba` will be divided into two different cakes `ab` and `ba`.

Can you help Ran figure out how many different pairs of cakes meet the above condition?

Input

The first line contains one integer n ($1 \le n \le 4 \cdot 10^5$) – the number of cakes.

The next n lines describe all the cakes, where the i-th line contains one string s_i ($1 \le |s_i| \le 4 \cdot 10^5$) consisting of lowercase Latin letters.

It is guaranteed that the sum of lengths of cakes does not exceed $4 \cdot 10^5$.

Output

Print one integer – the number of pairs of cakes meet the above condition.

Examples

standard input	standard output
3 ab ab cabc	3
3 abc a cabc	0
4 hhh hhh hhh hhh	6

Problem G. Grade Point Average

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

As soon as Moca entered the university last year as a freshman, she was told that the competition is fierce and everyone was desperately trying to improve their grades. As a result, in the last academic term, Moca studied just as hard as she did in high school.

The college will award scholarships based on the average score of the previous academic term. Unfortunately, someone has the same average score as Moca because the average score is calculated **rounding down** to only 1 decimal places. For example, Ran's average score is $\frac{48}{5} \approx 9.6$ and Moca's average score is $\frac{29}{3} \approx 9.6$, but apparently Moca's average score is higher.

Can you help Moca calculate her average score **rounding down** to k decimal places?

Input

The first line contains two integers n, k $(1 \le n, k \le 10^5)$ – the number of courses and the number of decimal places.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 100)$ – the score of each course.

Output

Print one real number in a line that denotes Moca's average score. Notice that your answer must contain k decimal places.

Example

standard input	standard output
3 10 94 100 99	97.6666666666

Problem H. Adventurer's Guild

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Yuna traveled into the fantasy world. The gods of this world gave her a powerful set of equipment so that she could defeat many fierce monsters. However, she had limited health points and stamina, and she could not kill a large number of monsters.

Adventurer's guild would release n monster crusade missions, such as black python and wild wolf. Completing the i-th mission would consume Yuna h_i health points and s_i stamina, and then she would get w_i gold coins.

In the beginning, Yuna had H health points and S stamina. When her health points were dropped to less than or equal to 0, she would die. However, when her stamina was dropped to less than 0, she would not give up, and then the overdrawn stamina would be reduced from health points. For example, her health points would be reduced by 3, when her stamina dropped to -3, and then her stamina would be reset to 0. If her health points can not afford the overdrawn stamina, she would also die.

As a friend of Yuna, can you help her choose some missions to complete to get the maximum number of gold coins? Make sure Yuna does not die, or you will be very sad.

Input

The first line contains three integers n, H, S $(1 \le n \le 1000, 1 \le H \le 300, 0 \le S \le 300)$.

The next n lines describe all the monster crusade missions, where the i-th line contains three integers h_i, s_i, w_i $(0 \le h_i, s_i \le 300, 1 \le w_i \le 10^9)$.

Output

Print one integer – the maximum number of gold coins that Yuna could get.

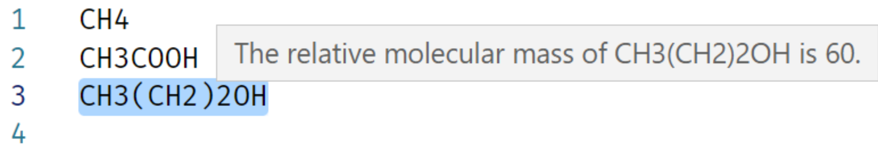
Examples

standard input	standard output
2 66 22 1 23 2 66 8 90	2
4 16 22 1 23 11 5 8 14 2 36 99 15 22 27	27

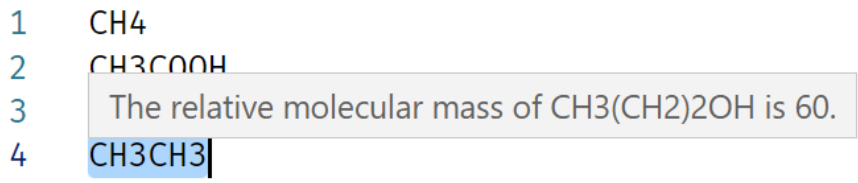
Problem I. Chemical Code

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

Chemical Code is an integrated development environment designed for chemists. Users can write the molecular formula in this editor, and if they move the mouse over the molecular formula, Chemical Code will display its relative molecular mass.



However, Moca, the developer of Chemical Code, did not implement this feature. Due to the tight schedule, she failed to meet the deadline, so that she just displayed the text "The relative molecular mass of CH3(CH2)2OH is 60.".



The molecular formula that Chemical Code supports consists of three types of symbols – chemical elements (`C`, `H`, `O`), digits and parentheses. **The relative molecular mass of `C` is 12. The relative molecular mass of `O` is 16. The relative molecular mass of `H` is** 1. The number written after a chemical element denotes that this element is repeated the corresponding number of times, and the number written after a pair of parentheses denotes where the subformula is also repeated the corresponding number of times. For example, `H2O` is expanded as `HHO`, and `(CH3)2` is expanded as `(CHHH)2` and `CHHHCHHH`. Notice that **nesting parentheses are allowed** and **multiple consecutive digits are not allowed**. The relative molecular mass of a molecular formula is the sum of the relative molecular mass of all the chemical elements after expanding this molecular formula. The following is the context-free grammar of the molecular formula, and the start symbol is E.

$\langle E \rangle$ ::= $\langle T \rangle \langle E \rangle \mid \langle T \rangle$

$\langle T \rangle$ ::= $\langle F \rangle \mid \langle F \rangle \langle D \rangle$

$\langle F \rangle$ ::= '`C`' | '`H`' | '`O`' | '`(`' $\langle E \rangle$ '`)`'

$\langle D \rangle$ ::= '`1`' | '`2`' | '`3`' | '`4`' | '`5`' | '`6`' | '`7`' | '`8`' | '`9`'

To simplify this feature, Chemical Code can be formed as a character array of length n, and it will handle the user's input operation. In the beginning, the array is initialized with n null characters. The user will perform q operations to fill the entire array. There are two types of the input operation. The first one is inputting a chemical element (`C`, `H`, `O`) or a digit in the specified position. And the second one is inputting a pair of parentheses in two specified positions. After each operation, Chemical Code will erase the null characters and print the relative molecular mass of the current molecular formula, which can be derived in the above context-free grammar. It is guaranteed that **each pair of parentheses input by the second**

type of operation is matching. For example, it is forbidden to input parentheses in the 2-nd and 5-th position in the molecular formula `(␣C)␣`.

Moca is asking for your help. Can you implement this feature for her?

Input

The first line contains two integers n, q $(1 \le q \le n \le 2 \cdot 10^5)$ – the length of the character array and the number of operations.

The next q lines describe the user's input operations, where the i-th line contains one of two operations:

- 1 c_i pos_i (c_i is `C`, `H`, `O` or a digit except 0, $1 \le pos_i \le n$) – user inputs a chemical element or a digit in the position pos_i;
- 2 L_i R_i $(1 \le L_i < R_i \le n)$ – user inputs a pair of parentheses in the position L_i and R_i.

It is guaranteed that the user does not input characters in a position twice, and after all the operations, there is no null characters in the character array.

Output

For each operation, print an integer in one line – the relative molecular mass of the current molecular formula. Since the answer may be very large, you only need to print the answer modulo 985151223.

Example

standard input	standard output
11 10 1 C 1 1 H 2 1 C 5 1 2 9 2 4 8 1 O 10 1 H 11 1 2 7 1 H 6 1 3 3	12 13 25 37 37 53 54 78 58 60

Note

The following table lists the answer of the above example.

Index	Molecular formula	Relative molecular mass
1	C	12
2	CH	13
3	CHC	25
4	CHC2	37
5	CH(C)2	37
6	CH(C)2O	53
7	CH(C)2OH	54
8	CH(C2)2OH	78
9	CH(CH2)2OH	58
10	CH3(CH2)2OH	60

Problem J. Tuition Agent

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Piggy is a selfless tuition agent. He already has n potential clients.

For the i-th client, Piggy must make one of two choices:

1. Spend X_i RMB training this client as a tutor;
2. Spend Y_i RMB making the client become a student who needs tutoring.

If Piggy can match a tutor with a student for one-on-one tutoring, he can earn K RMB in return. Of course, only those with higher rank can tutor those with lower rank. Note that 1 is the highest rank. The i-th client has a unique number R_i describing his grade rank, and no two clients have the same rank. Because of limited energy, each client is in at most one pair.

Now, Piggy wants to know the maximum profit he can get (the profit could be negative).

Input

The first line contains an integer T $(1 \le T \le 10)$ – the number of testcases.

For each testcase, the first line contains 2 integers n and k $(2 \le n \le 100000, 0 \le k \le 10000)$ – the number of clients and the money Piggy can collect.

The next n lines describe all the clients, where the i-th line contains three integers R_i, X_i, Y_i $(1 \le R_i \le n, 0 \le X_i, Y_i \le 10000)$.

It is guaranteed that testcases are generated randomly.

Output

For each test case, print the maximum profit Piggy can get.

Example

standard input	standard output
2 4 2 1 2 2 4 1 2 3 1 1 2 4 4 6 8 4 4 1 6 5 6 1 2 7 2 3 4 3 1 1 5 8 7	-5 4

Problem K. Piggy Calculator

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

Piggy is developing a fantastic calculator. This calculator innovatively invents a binary operator symbol that perfectly mixes addition and multiplication, which we call $\odot^k$ (k is a positive integer).

The value of $a \odot^k b$ is equal to $(a + b) \times k$. The precedence between operators is that the higher the k, the higher the priority. The higher priority goes first. For example, $\odot^7$ has a higher priority than $\odot^6$, so you must calculate $\odot^7$ before $\odot^6$. The operators are left-associative. Between the operators with the same priority, you should calculate them from left to right. For example, in the expression $1 \odot^1 2 \odot^1 3$, you must calculate $1 \odot^1 2$ firstly.

Now, Piggy wants to ask you to help him implement a function. There is an expression with n numbers and $n-1$ $\odot^k$ operators. At the same time, there are q queries, each of them shaped as (L, R) $(1 \le L \le R \le n)$. You need to calculate the result of the sub-expression formed by the L-th number to the R-th number.

Input

The first line contains one positive integer n $(2 \le n \le 3 \cdot 10^5)$ – the length of the expression.

The second line contains n positive integers $a_1, a_2, \ldots, a_n$ $(1 \le i \le n, 1 \le a_i \le 10^6)$ – the n numbers from left to right in the expression.

The third line contains $n - 1$ positive integers $k_1, k_2, \ldots, k_{n-1}$ $(1 \le i \le n - 1, 1 \le k_i \le 10^6)$, where the i-th operator from left to right is $\odot^{k_i}$ and is between the i-th number and the $i + 1$-th number.

The forth line contains one positive integer q $(1 \le q \le 3 \cdot 10^5)$ – the number of queris.

The next q lines describe the queries, where the i-th line contains two integers L_i, R_i $(1 \le L_i \le R_i \le n)$ that denote a query (L_i, R_i).

Output

For each query, print an integer in one line – the result of the calculation. Since the answer may be very large, you only need to print the answer modulo $10^9 + 7$.

Example

standard input	standard output
5 1 2 3 4 5 5 2 3 4 2 1 5 2 4	264 46

Note

In the first query, the equation is $1 \odot^5 2 \odot^2 3 \odot^3 4 \odot^4 5 = 15 \odot^2 3 \odot^3 4 \odot^4 5 = 15 \odot^2 3 \odot^3 36 = 15 \odot^2 117 = 264$.

In the second query, the equation is $2 \odot^2 3 \odot^3 4 = 2 \odot^2 21 = 46$.

Problem L. Construction of 5G Base Stations

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Construction of 5G base stations is underway on Piggy street. However, the service provider has run into some trouble, and he needs your help.

Piggy street can be abstracted as number axes. There are n residents living on integer points $1, 2, 3, 4, \ldots, n$. There are also m $(1 \le m \le n)$ 5G base stations located on the integer points of $1, 2, 3, 4, \ldots, n$, and **there will not be two base stations located at the same location**.

Every resident likes to solve programming problems on the network, so they need to establish a network connection with the base station. The probability of successful connection with a base station located at x_i is p_i $(1 \le i \le m)$.

The residents will try to connect to the base station from near to distant, and if the connection fails, the next one will be tried. If all the base stations have been tried, the cycle will be repeated – i.e., starting from the nearest base station until a connection is established. If there are two base stations at the same distance, the base station located to the left of the resident is preferred.

These base stations adopt an outdated and inefficient switching model. It connects two residents to communicate with each other through a physical line, which results in very high running costs. To be straightforward, suppose the number of residents connected to a particular base station is x, the running cost of that base station is x^2.

For the service provider, the cost of Piggy street is the sum of all the base station running costs. Now he asks you to tell him the cost expectation for Piggy street.

To avoid accuracy errors, it is guaranteed that the cost expectation can be represented as $\frac{P}{Q}$, where P and Q are coprime integers and $Q \not\equiv 0 (\bmod 998244353)$. Print the value of $P \cdot Q^{-1} \bmod 998244353$. Note that $p_1, p_2, \ldots, p_m$ are also given in the same form.

Input

The first line contains two positive integers n, m $(1 \le m \le n \le 10^6)$ – the length of the street and the number of base stations.

Each of the following m lines describes a base station with two integers x_i, p_i $(1 \le x_i \le n$, $0 < p_i < 998244353)$.

It is guaranteed that the probability that a resident fails to connect all the stations is not 1 modulo 998244353.

Output

Print one integer – the cost expectation modulo 998244353.

Examples

standard input	standard output
5 2 2 1 3 1	13
2 2 1 499122177 2 499122177	554580199

Note

In the first example, residents 1 and 2 connect to the first base station, and residents $3, 4$ and 5 connect to the second base station, so the running cost is $2^2 + 3^2 = 13$.

In the second example, the probability of successful connection with base stations 1 and 2 is $\frac{1}{2}$, and the cost expectation is $\frac{26}{9}$.

Problem M. Matrix Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

A teacher gives his students a problem to test his students' construction skills.

The teacher has two 0/1 matrices A and B with n rows and m columns, where all the 1 of both matrices separately form a 4-connected block. "Form one 4-connected block" means that starting at any 1, you can reach all the 1 by moving up, down, right or left, and you can not move into 0.

Now the teacher gives a third matrix C with n rows and m columns, and a specific position of this matrix is 1 if and only if the corresponding positions of the two matrices are 1.

You need to find two matrices A and B such that one of the positions of C is 1 if and only if the corresponding positions of A, B are both 1.

Input

The first line contains two integers n and m ($3 \le n, m \le 500$) – the size of the matrices.

The next n lines describe the matrix C. Each of the following n lines contains m characters 0 or 1.

It is guaranteed that the boundaries of the matrix C are 0, i.e., the 1-st row, the 1-st column, the n-th row and the m-th column of the matrix C are 0.

Output

Print $2n$ lines and each line has m characters 0 or 1.

The first n lines describe the matrix A.

The next n lines describe the matrix B.

If there are multiple answers, print any of them.

Example

standard input	standard output
5 5 00000 00100 01010 01100 00000	11110 10100 11110 11100 11110 00001 01111 01011 01111 00001