Problem A
Take It or Double It

Time limit: 1 second

Problem Description

You are the host of a street experiment show called **TOPC (Take Or Pass Challenge)**. In the show, you approach people one by one and offer them some money. Each person can either:

- **Take it**: Accept the money and end the game.
- **Double it**: Reject the money and ask you to double the amount and offer it to the next person.

The game starts with x dollars. You go down the street and meet *a random person*. You would like to predict in advance what each of them will say — either "take" or "double".

You have a limited budget of d dollars in your pocket. If, at any point, a person's decision could result in an amount **strictly greater** than your available funds d, you must **force the person to say "take it"**, regardless of what they wanted to say. Taiwanese people are very kind-hearted, so whenever they have a choice, they would always say "double it" and pass the opportunity to others — unless they are forced to take the money.

Your task is to simulate the game and determine if the person you meet will take the money or double it and give it to the next person.

Input Format

The input contains two integers x and d.

Output Format

Output a string "`take it`" or "`double it`".

Technical Specification

- $1 \leq x \leq d \leq 10^9$

Sample Input 1	Sample Output 1
`123 246`	`double it`

Sample Input 2	Sample Output 2
`345 678`	`take it`

Sample Input 3

```
789 101112
```

Sample Output 3

```
double it
```

Problem B
Twin Guardians

Time limit: 1 second

Problem Description

In a distant Mathematical Kingdom, there stands an ancient gate known as the "Twin Gate." Legend has it that the gate will only open when two "Twin Guardians" stand before it at the same time.

The king has summoned a wise adventurer to help determine whether the two given numbers are "Twin Primes." If they genuinely are Twin Guardians (i.e., Twin Primes), the gate will shine brightly and open; otherwise, it will remain firmly shut.

Definition of Twin Primes: If two numbers i and $i+2$ are both prime, they are called "twin primes."

Input Format

Each test contains multiple test cases. The first line contains the number of test cases t. The description of the test cases follows.

The only line of each test case contains two integers a and b, representing a task assigned by the king that asks you to check these two numbers.

Output Format

For each test case, if a and b are twin primes, output `Y` (indicating "Yes, the Twin Gate opens!"); otherwise, output `N`.

Technical Specification

- $1 \leq t \leq 10$
- $1 \leq a < b \leq 10^6$

Sample Input 1

```
5
2 3
11 13
12 14
3 5
5 7
```

Sample Output 1

```
N
Y
N
Y
Y
```

Sample Input 2

```
5
17 19
29 31
15 17
19 21
41 43
```

Sample Output 2

```
Y
Y
N
N
Y
```

Problem C
One-Way Abyss
Time limit: 1 second

Problem Description

Mitty is a brave adventurer exploring a mysterious underground cave system known as *The Abyss*. The Abyss is composed of n parallel vertical shafts and m horizontal tunnels. Each tunnel connects exactly two shafts at the same depth. All m tunnels have distinct depths, and surprisingly, there is a treasure in the middle of every tunnel!

Mitty can pick any shaft to start with. He moves downward from the top of the chosen shaft, obeying the following rules:

- He can only move downward, going upward is not allowed.
- Whenever he encounters a horizontal tunnel, he **must** enter it immediately and will arrive at the connected shaft.
- Once he enters a horizontal tunnel, he cannot go back.

These treasures in the tunnels have various values. Mitty wants to collect as much treasure as possible. Please help him calculate the maximum total value of treasures he can collect when starting from one of the shafts.

Input Format

Each test contains multiple test cases. The first line contains the number of test cases t. The description of the test cases follows.

The first line contains two integers n and m, representing the number of vertical shafts and horizontal tunnels, respectively.

Each of the following m lines contains three integers x, y and v, representing a horizontal tunnel at a certain depth that connects shafts numbered x and y, and contains a treasure worth v.

The horizontal tunnels are given from top to bottom. This implies that no two horizontal tunnels situated at the same depth.

Output Format

For each test case, print a single integer, representing the maximum total value of treasures Mitty can collect.

Technical Specification

- $1 \le t \le 20$
- $1 \le n \le 2 \times 10^5$
- $0 \le m \le 2 \times 10^5$
- $1 \le x < y \le n$
- $0 \le v \le 10^9$
- It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.
- It is guaranteed that the sum of m over all test cases does not exceed 2×10^5.

Sample Input 1

```
1
3 3
1 2 3
2 3 4
1 3 9
```

Sample Output 1

```
16
```

Sample Input 2

```
2
5 8
1 4 5
1 3 4
1 3 2
1 3 9
2 4 1
1 3 2
2 3 0
1 5 6
7 2
5 6 16
5 7 4
```

Sample Output 2

```
28
20
```

Problem D
Palindromic Distance

Time limit: 1 second

Problem Description

The edit distance between two strings u and v is the minimum number of edit operations that turns u into v. There are three edit operations that can be applied to a string: Insert a character, delete a character, and substitute some character by a different one.

For example, we can turn `hello` into `world` with four substitutions, so the edit distance is at most 4. You can turn `wally` into `walter` with two substitutions and one insertion, so the edit distance is at most 3. Computing the edit distance between two strings is a well-known problem with many applications.

The task at hand is to compute the edit distance of a string to **one of the closest** palindromes. A palindrom is a string that is the same when read backwords, for example `madam`.

The edit distance of `hello` to the closest palindrome is only 2: We can turn `hello` into `ollo`, or `hllh`, or `elle` with two edit operations.

Write a program that can find the distance of a word to a closest palindrome.

Input Format

Each test contains multiple test cases. The first line contains the number of test cases t. The description of the test cases follows.

The only line of each test case contains a word w consisting only of lowercase letters.

Output Format

For each test case, output the edit distance of the input word w to its closest palindrome.

Technical Specification

- $1 \leq t \leq 200$
- The word w has length at least one.
- It is guaranteed that the sum of the lengths of the words w over all test cases does not exceed 3000.

Sample Input 1

```
6
aaaaba
hello
palindrome
abba
x
bababac
```

Sample Output 1

```
1
2
5
0
0
1
```

Problem E
Explosive Slabstones Rearrangement

Time limit: 1 second

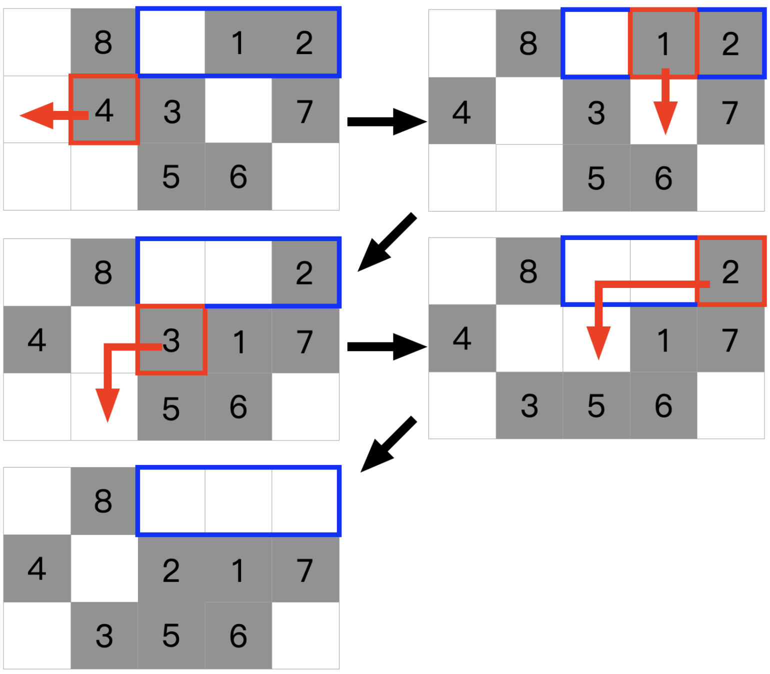
Problem Description

Barbara has a garden. The garden can be represented as a $n \times m$ grid. She has placed k slabstones in the garden, so she can enjoy stepping slabstones from one to another every day. The slabstones are indexed from 1 to k. Each slabstone fully occupies some cell of the garden, and no two slabstones are placed at the same cell.

However, an evil person, Babara, is going to place a special device that will occupy a rectangular region in the garden. If any slabstone overlaps with the device, it will explode and destroy the whole garden!

To avoid the explosion, Barbara wants to rearrange the slabstones by shifting the slabstones within the garden. The slabstones should remain non-overlapping **during slabstone rearrangement**. The energy spent is equal to the largest index among all slabstones that have been moved. Now Barbara wants to know the minimum energy required to rearrange the slabstones, so she can save her energy for "other purposes".

For example, suppose the device will be placed at the blue rectangle. Then Barbara can rearrange the slabstones as follows. Please note that the slabstones do not overlap during the whole process, not only after the rearrangement. All slabstones that have been moved have index at most 4, so the energy spent is 4.



Input Format

The first line contains three integers n, m and k.

Followed by k lines, the i-th of which contains two integers x_i and y_i, representing that the i-th slabstone is located at the y_i-th cell of the x_i-th row.

The last line contains four integers u_1, v_1, u_2 and v_2, representing that the top-left corner of the device is located at the v_1-th cell of the u_1-th row, and the bottom-right corner of the device is located at the v_2-th cell of the u_2-th row.

Output Format

Print the minimum energy required to rearrange the slabstones. If no slabstones need to be moved, the energy spent is 0. If the explosion cannot be avoided, just output -1.

Technical Specification

- $1 \le n \le 500$
- $1 \le m \le 500$
- $1 \le k \le n \times m$
- $1 \le x_i \le n$
- $1 \le y_i \le m$
- $1 \le u_1 \le u_2 \le n$
- $1 \le v_1 \le v_2 \le m$
- All (x_i, y_i) pairs are distinct.

Sample Input 1

```
3 5 8
1 4
1 5
2 3
2 2
3 3
3 4
2 5
1 2
1 3 1 5
```

Sample Output 1

```
4
```

Sample Input 2

```
3 3 1
2 2
1 1 3 3
```

Sample Output 2

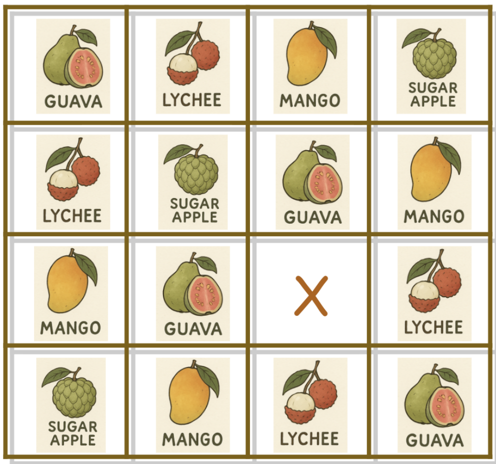
```
-1
```

This page is intentionally left blank.

Problem F
Fruitful Compression

Time limit: 3 seconds

Problem Description



Generated by ChatGPT 4o

Alice and Bob run a fruit business that delivers delicious fruits to customers around the world. Their brand showcases four iconic fruits from Taiwan: guava, lychee, mango, and sugar apple.

They've opened multiple branches, and each branch uses the same layout to display fruits. Each day, the storefront of every shop is arranged as a 4×4 grid. The grid displays fruits such that every row and every column contains exactly one of each of the four different fruits—no duplicates per row or column.

To ensure consistency across branches, every evening, Alice and Bob prepare a 4×4 fruit box with a valid display layout and have their employees deliver the boxes to each branch the next morning.

Alice and Bob have two children, Charlie and Dianne, who love eating fruit. One evening, they sneak into the office and find the prepared fruit boxes sitting on the table.

Charlie can't resist—he grabs one fruit and eats it. "Mmm... I wish I could eat as many as possible without confusing the staff tomorrow," he thinks.

Dianne, noticing her brother's mischief and feeling hungry herself, takes another fruit. Soon, the two begin taking turns, each grabbing one fruit at a time from the box.

Their rule is simple: They can only take a fruit if, after removing it, the remaining fruits in the box still uniquely determine the original 4×4 display (i.e., there's only one way to complete the display while preserving the distinct-fruit-per-row-and-column rule).

Neither Charlie nor Dianne wants to be the one who can't grab a fruit anymore. So they both follow the same strategy:

"If there's a way I can eventually grab the last fruit without violating the rule, I'll take it, and I'll try to maximize the total number of fruits I can eat. But if my sibling could outsmart me and grab the last fruit, I'll instead try to minimize the number of fruits they can take."

Now, you are given the current status of the fruit box. That is, some fruits are already taken under the rule. Your task is to write a program that predicts the total number of fruits that will remain in the box after both kids have played optimally.

Input Format

Each test contains multiple test cases. The first line contains the number of test cases t. The description of the test cases follows.

Each test case contains four lines, the i-th of which contains a string of length four, representing the i-th row of the layout of the fruit box. Each character will be in the set {`G`, `L`, `M`, `S`, `X`}, indicating each of the four fruits, or an empty slot denoted by `X`.

Output Format

For each test case, please output the specified answer in each line.

Technical Specification

- $1 \leq t \leq 10$
- It is guaranteed that each input fruit box is valid. That is, there exists a unique way to fill in the empty slots such that each row and each column has distinct fruits.

Sample Input 1

```
2
GLMS
LSGM
MGXL
SMLG
GLMS
LMSG
MSXL
SGLM
```

Sample Output 1

```
5
5
```

Problem G
Gamer Bafuko

Time limit: 2 seconds

Problem Description

Bafuko loves gaming! She is currently playing the latest title, *Young Bracelet – Light of the Saint Grass*. As an experienced gamer, she wants to explore every location on the map.

Picture of Bafuko for reference (X @bafuko_seiso)

The map can be represented as a tree with n vertices numbered from 1 to n. Each edge of the tree has weight 1. In addition, there is a special portal connecting two designated distinct vertices x and y. This portal can be used any number of times and allows instantaneous travel between x and y at zero cost.

Bafuko wants to visit every vertex of the tree at least once by following a tour. A tour is a sequence of adjacent vertices that begins at a chosen vertex, ends at a chosen vertex (possibly the same one), and visits every vertex of the tree at least once. She may freely choose the starting and ending vertices. During the tour, both edges and the portal may be traversed multiple times if necessary. The cost of the tour is defined as the sum of the costs of all traversed edges and portal uses, where repeated traversals are counted multiple times.

Your task is to determine the minimum total cost of such a tour.

Input Format

Each test contains multiple test cases. The first line contains the number of test cases t. The description of the test cases follows.

The first line of each test case contains three integers n, x, and y, representing the number of vertices in the tree and the two vertices connected by the portal.

Followed by $n - 1$ lines, the i-th of which contains two integers u_i, v_i, representing an edge between vertices u_i and v_i.

Output Format

For each test case, print a single integer in a new line, representing the minimum total cost of a tour that visits every vertex of the tree at least once.

Technical Specification

- $1 \le t \le 10^4$
- $2 \le n \le 5 \times 10^5$
- $1 \le x, y \le n$ and $x \ne y$.
- $1 \le u_i, v_i \le n$ and $u_i \ne v_i$.
- It is guaranteed that the given edges form a tree.
- It is guaranteed that the sum of n over all test cases does not exceed 5×10^5.

Sample Input 1

```
3
4 1 3
1 2
2 3
2 4
8 1 3
1 2
2 3
2 4
4 7
7 8
4 5
5 6
8 1 2
1 2
2 3
3 4
4 5
3 6
6 7
7 8
```

Sample Output 1

```
2
8
7
```

Problem H
Chopsticks

Time limit: 2 seconds

Problem Description

Chisato works at a traditional Japanese restaurant that just received a shipment of beautifully handcrafted chopsticks. There are m different types of chopsticks, and for each type i ($1 \leq i \leq m$), there are exactly k_i chopsticks.

Tonight, n guests have arrived, and each guest needs exactly one pair of chopsticks. Since no type of chopstick has at least $2n$ pieces, Chisato decides to randomly select $2n$ chopsticks from the full collection, which contains $s = \sum_{i=1}^{m} k_i$ chopsticks in total.

After selecting the $2n$ chopsticks, Chisato will try to distribute them in a way that maximizes the number of guests receiving a matching pair, that is, two chopsticks of the same type. If it's not possible to provide matching pairs for everyone, some guests will receive mismatched pairs.

Your task is to compute the expected number of guests who receive mismatched pairs of chopsticks under this strategy.

Input Format

The first line contains two integers n and m, representing the number of people and the number of the chopstick type, respectively.

The second line contains m integers, the i-th integer k_i represents the number of chopstick for the i-th type.

Output Format

Print a single integer, the expected number of people who cannot get a pair of chopsticks of the same type, multiplied by $\binom{s}{2n}$ (where $s = \sum_{i=1}^{m} k_i$). It can be proven that this product is an integer. Output the result modulo 998244353.

Technical Specification

- $1 \leq n \leq 2.5 \times 10^5$
- $1 \leq m \leq 5 \times 10^5$
- $1 \leq k_i < 2 \times n$
- $2 \times n \leq s$

Sample Input 1

```
3 3
2 2 2
```

Sample Output 1

```
0
```

Sample Input 2

```
5 3
3 3 4
```

Sample Output 2

```
1
```

Sample Input 3

```
5 2
8 8
```

Sample Output 3

```
4032
```

Problem I
Reactor

Time limit: 7 seconds

Problem Description

In a high-tech industrial facility, a series of nuclear reactors are arranged in a linear configuration. Each reactor operates under strict pressure regulations to ensure safety and efficiency. To prevent critical failures, each reactor has a specific maximum pressure limit. When a reactor's internal pressure reaches or exceeds this limit, a controlled pressure release (venting) is initiated. This system requires sophisticated management due to dynamic operational adjustments and the need for continuous monitoring.

You are tasked with designing and implementing a system to manage the pressure of a line of n reactors. Each reactor, indexed from 1 to n, has an initial maximum pressure limit p_i. All of the reactors' initial pressure are 0. The system must support two types of operations:

1. **Pressure Increase Operation:** For a given range of reactors $[l, r]$, increase their pressure by k units. If the pressure of any reactor in this range reaches or exceeds its maximum limit, it will vent, resetting its pressure to 0. And the maximum pressure limit of the vented reactor will be updated to $\max(\lfloor \frac{p_{old}}{2} \rfloor, 1)$, where p_{old} is the maximum pressure limit of the reactor before the current pressure increase operation.
2. **Venting Count Query:** For a given range of reactors $[l, r]$, you need to report the total number of venting operations that have occurred among all reactors within this specified range since the beginning of the system's operation.

Input Format

The first line contains two integers n and q, representing the number of reactors and the number of operations, respectively.

The second line contains n integers, the i-th integer p_i represents the initial maximum pressure limit of the i-th reactor.

The following q lines describe the operations. Each line begins with an integer op.

- If $op = 1$, it is followed by three integers l, r, and k, representing a pressure increase operation on the range of reactors from l to r (inclusive) by k units.
- If $op = 2$, it is followed by two integers l and r, representing a venting count query for the range of reactors from l to r (inclusive).

Output Format

For each query that $op = 2$, print a single integer on a new line, representing the total number of venting operations that have occurred among all reactors within the specified range since the beginning of the system's operation.

Technical Specification

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq q \leq 2 \times 10^5$
- $1 \leq p_i \leq 4 \times 10^5$
- $1 \leq l \leq r \leq n$
- $1 \leq k \leq 4 \times 10^5$
- It is guaranteed that there is at least one Venting Count Query.

Sample Input 1

```
10 5
5 10 23 45 10 45 65 10 68 9
1 5 10 664
1 2 9 5
2 4 10
1 8 8 5
2 1 10
```

Sample Output 1

```
8
9
```

Sample Input 2

```
10 10
79 26 9 28 13 40 26 54 69 19
1 1 5 6
1 5 7 2
2 4 7
1 9 10 19
2 5 7
1 5 7 27
2 10 10
2 9 9
1 6 6 20
1 3 8 6
```

Sample Output 2

```
0
0
1
0
```

Sample Input 3

```
10 10
56 29 49 42 47 21 23 54 8 31
2 9 9
1 5 6 23
2 6 7
2 4 7
1 5 6 68
2 1 9
2 3 6
1 2 10 89
2 6 8
1 3 6 53
```

Sample Output 3

```
0
1
1
3
3
5
```

This page is intentionally left blank.

Problem J
Gas Station

Time limit: 3 seconds

Problem Description

Alex is planning rest area placements on a simplified model of Taiwan's freeway system. The system contains n interchanges, connected by $n-1$ bidirectional roads. The network is connected, and there is exactly one shortest route between any pair of interchanges. The i-th road connects interchanges u_i and v_i, and has a length of l_i.

Exactly k rest areas with gas stations can be built, each located at an interchange. A driver may start a trip from any interchange and travel to any other, always following the unique shortest path. They begin each trip with a full tank of gas and can refuel only at interchanges that have a rest area.

Alex is curious about the smallest possible fuel tank capacity d such that it's possible to place the k rest areas in a way that ensures no driver will ever run out of gas. On any trip, the driver must never have to travel more than d units along the path without passing through a rest area, including at the beginning or end of the journey. The goal is to figure out the minimum such d, assuming the rest areas are placed in the best possible way.

Input Format

The first line contains two integer n, k.

Followed by $n-1$ lines, the i-th of which contains three integers u_i, v_i, l_i, representing the i-th road connects interchanges u_i and v_i with a length l_i.

Output Format

Output one integer, the smallest possible fuel tank capacity d.

Technical Specification

- $2 \leq n \leq 2 \times 10^5$
- $0 \leq k \leq n$
- $1 \leq u_i, v_i \leq n$
- $1 \leq l_i \leq 10^9$
- It is guaranteed that the input roads form a tree.

Sample Input 1

```
5 1
1 2 3
1 5 2
2 3 3
2 4 1
```

Sample Output 1

```
5
```

Sample Input 2

```
5 2
1 2 3
1 5 2
2 3 3
2 4 1
```

Sample Output 2

```
3
```

Problem K
Move Stone

Time limit: 1 second

Problem Description

You are given an $n \times n$ grid. Each cell initially contains some number of stones, such that the total number of stones is exactly n^2.

In one move, you may take a single stone and move it to any other cell in the same row or the same column.

Your goal is to minimize the number of moves needed to make each cell contain exactly one stone.

Input Format

The first line contains an integer n, representing the size of the grid.

Followed by n lines, the i-th of which contains n integers, the j-th integer $a_{i,j}$ represents the number of stones in cell (i, j).

Output Format

Output a single integer, the minimum number of moves required to make each cell contain exactly one stone.

Technical Specification

- $1 \leq n \leq 500$
- $0 \leq a_{i,j} \leq n^2$
- The initial number of stones is exactly equal to the number of cells on the board.

Sample Input 1

```
3
0 1 2
0 2 2
1 1 0
```

Sample Output 1

```
3
```

Sample Input 2

```
5
1 2 4 0 1
2 0 0 2 0
1 4 1 0 1
2 0 0 0 0
1 2 0 1 0
```

Sample Output 2

```
11
```

Problem L
Stapler

Time limit: 2 seconds

Problem Description

Alice is selling a new model of game console in the toy store. When selling a new console, Alice uses a stapler to attach the receipt to the box.

However, the stapler pierces the box at two points, causing damage along the entire line segment between those two points. If any part of this segment (including its endpoints) overlaps with the console's screen (including its boundary), it will be damaged and Alice must compensate for the console.

Fortunately, a laser reveals the exact position of the screen inside the box. The game console's screen is a rectangle with sides parallel to the coordinate axes. Its bottom-left corner is at (x_l, y_l) and its top-right corner is at (x_r, y_r).

Alice plans to staple the box at the points (x_1, y_1) and (x_2, y_2). Please help her determine whether the stapler will damage the screen. If it will, stop her immediately.

Input Format

Each test contains multiple test cases. The first line contains the number of test cases t. The description of the test cases follows.

The first line of each test case contains four integers x_l, y_l, x_r, y_r, representing the coordinates of the bottom-left and top-right corners of the screen.

The second line of each test case contains four integers x_1, y_1, x_2, y_2, representing the coordinates of the expected stapler penetration positions.

Output Format

For each test case, print `STOP` on a line if Alice will damage the screen, and print `OK` otherwise.

Technical Specification

- $1 \leq t \leq 10^4$
- The area of the screen is greater than 0.
- The length of the stapler is greater than 0.
- All coordinates are integers with absolute value at most 10^4.

Sample Input 1

```
3
0 0 5 5
0 10 10 0
0 0 5 5
6 6 6 7
0 0 5 5
1 1 2 2
```

Sample Output 1

```
STOP
OK
STOP
```