

UCPC 2026 - II stage

Ukrainian Collegiate Programming Contest

October 24, 2026

Problems

A Archive Tree
Time 1 s · Memory 256 MiB

B Bitwise Potholes
Time 4 s · Memory 256 MiB

C Convenient Shipments
Time 9 s · Memory 256 MiB

D Days of the Week
Time 0.5 s · Memory 256 MiB

E Exam Session
Time 1 s · Memory 256 MiB

F Four Steps
Time 2.5 s · Memory 256 MiB

G Grouped Vertices
Time 3 s · Memory 256 MiB

H Hall of Songs
Time 1 s · Memory 256 MiB

I Inconvenient Swaps
Time 1 s · Memory 256 MiB

J Jar of Colours
Time 1.5 s · Memory 256 MiB

K Know This Problem?
Time 1 s · Memory 256 MiB

L Lost List
Time 1 s · Memory 256 MiB

M Mosaic Arch
Time 1 s · Memory 256 MiB

A Archive Tree

Time limit 1 s · Memory limit 256 MiB · Standard input/output

An archive consists of n folders numbered from 1 to n. Some pairs of folders are directly linked, there are $n-1$ such links in total, and from any folder one can reach any other one through the links; in other words, the folders and the links form a tree. Folder s is the *root* folder.

For every folder $v \neq s$, its *parent* folder is the neighbour of v on the path from v to the root folder, and v is called a *subfolder* of its parent.

Every folder v must be given a name f_v, which is a positive integer. By the convention of the archive, the names must satisfy the following rules:

- the root folder is named $f_s = 1$;
- the name of every other folder v is a multiple of the name of its parent folder p, that is, $f_v = k \cdot f_p$ for some positive integer k (the case $k = 1$ is allowed: a subfolder may have the same name as its parent);
- two subfolders of the same folder must have different names.

Short names are easier to type, so the administrator wants the largest name in the archive to be as small as possible. Find the minimum possible value of $\max\limits_{1 \le v \le n} f_v$.

Input

The first line contains a single integer t ($1 \le t \le 5 \cdot 10^5$) — the number of test cases.

The first line of each test case contains two integers n and s ($1 \le s \le n \le 5 \cdot 10^5$) — the number of folders and the number of the root folder. Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n$, $u \neq v$) — the folders connected by a link. It is guaranteed that the links form a tree.

The sum of n over all test cases does not exceed $5 \cdot 10^5$.

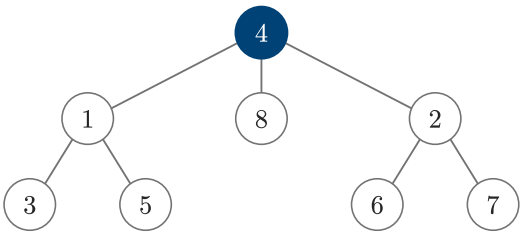
Output

For each test case print one integer — the minimum possible largest name.

Examples

Input	Output
3 8 4 4 1 4 2 8 4 1 3 5 1 2 6 7 2 5 3 1 2 3 2 3 4 5 4 1 1	4 2 1

Note



In the first test case the root folder is 4, and it has the subfolders 1, 2 and 8. Folder 1 has the subfolders 3 and 5, and folder 2 has the subfolders 6 and 7. One optimal assignment of names is:

- $f_4 = 1$;
- $f_1 = 1$, $f_2 = 2$, $f_8 = 3$;
- $f_3 = 1$, $f_5 = 2$;
- $f_6 = 2$, $f_7 = 4$.

It can be shown that the largest name cannot be less than 4.



In the second test case the folders form the chain $1 - 2 - 3 - 4 - 5$ with the root folder 3 in the middle. Folders 2 and 4 are subfolders of the root folder, so they need different names. For example, $f_1 = f_2 = f_3 = 1$ and $f_4 = f_5 = 2$, so the answer is 2.

In the third test case there is only the root folder, and $f_1 = 1$.

B Bitwise Potholes

Time limit 4 s · Memory limit 256 MiB · Standard input/output

A country has n cities and $n-1$ two-way roads, each connecting two different cities. The road network is *connected*: one can get from any city to any other one. Hence any two cities are joined by exactly one *route*, a path that visits no city twice.

Every road has a non-negative integer number of potholes. The president wants to make every route between two different cities smooth. A route is called *smooth* if the bitwise AND of the numbers of potholes on its roads is equal to their bitwise OR (the bitwise operations are recalled below).

A road crew can repeat the following operation any number of times:

- Choose two different cities u and v and an integer $i \geq 0$, and on every road of the route from u to v replace the number of potholes w by $w \oplus 2^i$ (that is, flip bit i).

Recall the definitions. *Bit number* i of a non-negative integer is its binary digit at the position of 2^i (bit 0 is the least significant one).

- The *bitwise AND* of several integers has bit i equal to 1 if and only if bit i equals 1 in *every* one of them.
- The *bitwise OR* of several integers has bit i equal to 1 if and only if bit i equals 1 in *at least one* of them.
- The *bitwise XOR* ($\oplus$) of two integers has bit i equal to 1 if and only if bit i differs in these two integers. In particular, $w \oplus 2^i$ is w with bit i flipped.

Find the **minimum** number of operations needed to make **every route** between two different cities smooth.

Input

The first line contains a single integer t ($1 \leq t \leq 10^5$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of cities. Each of the next $n-1$ lines contains three integers u, v and w ($1 \leq u, v \leq n$, $u \neq v$, $0 \leq w < 2^{30}$): a road between cities u and v with w potholes. It is guaranteed that the road network is connected.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case print one integer — the minimum number of operations.

Examples

Input

```
5
5
1 2 1
3 2 0
3 4 0
5 4 1
5
3 1 6
2 3 6
3 4 6
5 3 2
1
4
1 2 3
3 2 5
4 3 6
7
1 2 0
2 3 1
1 4 0
4 5 1
1 6 0
6 7 1
```

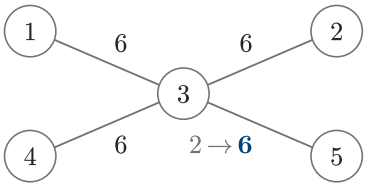
Output

```
1
1
0
3
2
```

Note

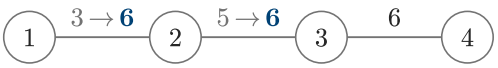


In the first test case the cities form a chain $1-2-3-4-5$ with $1,0,0,1$ potholes on its roads. One operation is enough: flip bit 0 on the route from city 2 to city 4. Then every road has 1 pothole.

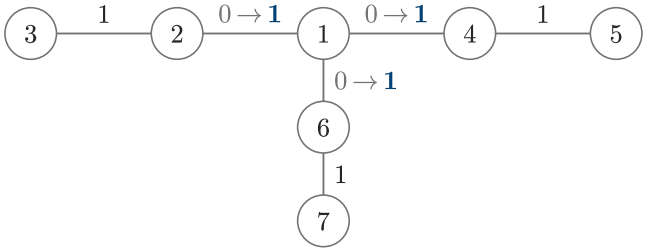


In the second test case one operation on the route from city 3 to city 5 with $i=2$ changes 2 into 6, and all four roads have 6 potholes.

In the third test case there is only one city, so there are no routes between different cities and nothing needs to be done.



In the fourth test case the chain $1-2-3-4$ has $3,5,6$ potholes. Three operations are enough: route $1 \to 3$ with $i=0$, route $2 \to 3$ with $i=1$ and route $1 \to 2$ with $i=2$. Then every road has 6 potholes.



In the fifth test case city 1 is joined to cities 2, 4 and 6, and the roads $2-3$, $4-5$ and $6-7$ continue the three branches. The three roads at city 1 have 0 potholes, and the outer three have 1. Two operations are enough: route $2 \to 4$ with $i = 0$ and route $1 \to 6$ with $i = 0$. Then every road has 1 pothole. It can be shown that one operation is not enough.

C Convenient Shipments

Time limit 9 s · Memory limit 256 MiB · Standard input/output

A train of n wagons stands on the main track of a freight station. The wagons are numbered from 1 to n from the locomotive, and the i-th wagon carries a_i identical boxes.

The station sends cargo off in shipments. A shipment is a group of consecutive wagons $i, i+1, \dots, j$, and all its boxes are packed into containers. All containers used for one shipment must have the same capacity $d \geq 2$, and every wagon must be packed completely, without leftovers. A group of consecutive wagons is called *convenient* if it can be sent as one shipment, that is, if there is an integer $d \geq 2$ such that the number of boxes in each wagon of the group is divisible by d. In other words, the group is convenient if and only if $\gcd(a_i, a_{i+1}, \dots, a_j) > 1$. A single wagon with $a_i \geq 2$ boxes forms a convenient group; a wagon with exactly one box does not.

During the day q messages of two kinds arrive:

- `1 p x` — wagon p has been reloaded and now carries x boxes, that is, a_p becomes equal to x;
- `2 l r` — the dispatcher wants to know how many convenient groups consist only of the wagons $l, l+1, \dots, r$, that is, the number of pairs (i, j) with $l \leq i \leq j \leq r$ such that $\gcd(a_i, a_{i+1}, \dots, a_j) > 1$.

Answer all messages of the second kind.

Input

The first line contains an integer t ($1 \leq t \leq 3 \cdot 10^5$) — the number of test cases.

The first line of each test case contains two integers n and q ($1 \leq n, q \leq 3 \cdot 10^5$) — the number of wagons and the number of messages. The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — the initial numbers of boxes.

Each of the next q lines contains a message in one of the two forms:

- `1 p x` ($1 \leq p \leq n$, $1 \leq x \leq 10^9$);
- `2 l r` ($1 \leq l \leq r \leq n$).

The sum of n over all test cases does not exceed $3 \cdot 10^5$, and the sum of q over all test cases does not exceed $3 \cdot 10^5$.

Output

For each message of the second kind print one integer on a separate line — the number of convenient groups inside the given segment. Print the answers in the order in which the messages are given, test case after test case.

Examples

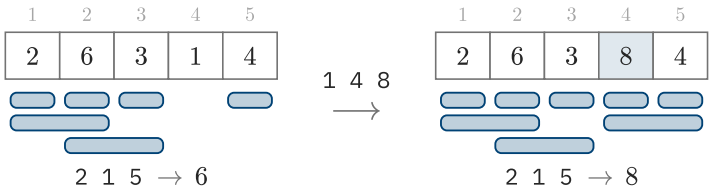
Input

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

Output

```
6
3
8
4
6
2
0
```

Note



The figure shows the first test case: every convenient group for the two messages `2 1 5` is drawn as a bar under its wagons.

In the first test case the train is initially $2, 6, 3, 1, 4$. A group of wagons from i to j is written as $[i, j]$.

- Message `2 1 5`: the convenient groups are $[1, 1]$, $[1, 2]$, $[2, 2]$, $[2, 3]$, $[3, 3]$, $[5, 5]$, so the answer is 6. For example, $[1, 3]$ is not convenient because $\gcd(2, 6, 3) = 1$, and $[4, 4]$ is not convenient because wagon 4 carries one box.
- Message `2 2 3`: the convenient groups are $[2, 2]$, $[2, 3]$, $[3, 3]$, so the answer is 3.
- Message `1 4 8`: the train becomes $2, 6, 3, 8, 4$.
- Message `2 1 5`: the convenient groups are $[1, 1]$, $[1, 2]$, $[2, 2]$, $[2, 3]$, $[3, 3]$, $[4, 4]$, $[4, 5]$, $[5, 5]$, so the answer is 8.
- Message `2 3 5`: the convenient groups are $[3, 3]$, $[4, 4]$, $[4, 5]$, $[5, 5]$, so the answer is 4.

In the second test case the train is $2, 2, 3, 3$.

- Message `2 1 4`: the convenient groups are $[1, 1]$, $[1, 2]$, $[2, 2]$, $[3, 3]$, $[3, 4]$, $[4, 4]$, so the answer is 6. No group containing both wagons 2 and 3 is convenient, since $\gcd(2, 3) = 1$.
- Message `2 2 3`: the convenient groups are $[2, 2]$ and $[3, 3]$, so the answer is 2.

In the third test case the train is 1.

- Message `2 1 1`: the only wagon carries one box, so there are no convenient groups and the answer is 0.

D Days of the Week

Time limit 0.5 s · Memory limit 256 MiB · Standard input/output

Every autumn Ukrainian student teams go through the ICPC season: first the 1/8 stage, then the 1/4 stage, and finally SEERC. This year the three stages took place on three different calendar days: the 1/8 stage was held before the 1/4 stage, and the 1/4 stage before SEERC.

Olena does not remember the dates any more, only the day of the week of each stage. The days of the week repeat in the order `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, `sunday`, then `monday` again, and so on.

Help Olena find the minimum possible number of days from the 1/8 stage to SEERC, counting both of these days, that is, the number of days on which the season had already started and had not yet finished.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases.

Each of the next t lines contains three words: the days of the week of the 1/8 stage, of the 1/4 stage and of SEERC, in this order. Each word is one of `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, `sunday`.

Output

For each test case print one integer — the minimum possible number of days from the 1/8 stage to SEERC inclusive.

Examples

Input	Output
3 monday monday monday sunday monday tuesday friday tuesday saturday	15 3 9

Note

In the first test case the stages could take place on three Mondays in a row, for example on October 5, October 12 and October 19: that is 15 days from the first to the last one inclusive.

In the second test case the stages could have taken place on three consecutive days.

In the third test case the 1/4 stage was at least 4 days after the 1/8 stage (from a Friday to the next Tuesday), and SEERC was at least 4 days after the 1/4 stage (from a Tuesday to the next Saturday), so the answer is $1 + 4 + 4 = 9$.

E Exam Session

Time limit 1 s · Memory limit 256 MiB · Standard input/output

Oksana is a student of the Faculty of Cybernetics, and her exam session is unusually flexible this year. She has to pass n exams. The days of the session are numbered with positive integers, and the i-th exam can be taken on any day from day l_i to day r_i inclusive.

Oksana does not want to overload herself, so she is going to take **at most one exam per day**. Each exam has to be taken exactly once.

Determine whether Oksana can choose a day for every exam so that she passes all n exams. Formally, check whether there exist integers $x_1, x_2, \ldots, x_n$ such that $l_i \le x_i \le r_i$ for every i and all x_i are pairwise distinct.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

The first line of each test case contains an integer n ($1 \le n \le 2 \cdot 10^5$) — the number of exams. Each of the next n lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le 10^9$) — the first and the last day on which the i-th exam can be taken.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

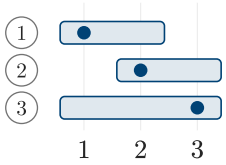
For each test case print `YES` if Oksana can pass all exams, and `NO` otherwise.

You may print each letter in any case (for example, `yes`, `Yes` and `YES` are all accepted).

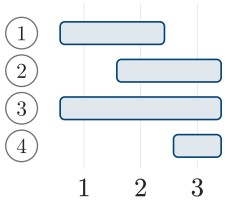
Examples

Input	Output
3 3 1 2 2 3 1 3 4 1 2 2 3 1 3 3 3 2 5 5 1 1000000000	YES NO YES

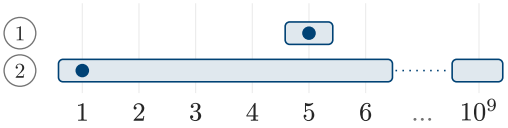
Note



In the first test case Oksana can take the exams on days 1, 2 and 3, respectively.



In the second test case all four exams must be taken on days from 1 to 3, which is impossible.



In the third test case she can take the first exam on day 5 and the second exam on any other day, for example, on day 1.

F Four Steps

Time limit 2.5 s · Memory limit 256 MiB · Standard input/output

You are given a string s of length n consisting of the characters `0`, `1`, `2`, `3` and `?`. Every character `?` must be replaced by one of the digits 0, 1, 2, 3 (different occurrences may be replaced by different digits).

For a string $u_1 u_2 \ldots u_n$ of digits, a *climb* is a quadruple of indices $i < j < k < l$ such that $u_i = 0$, $u_j = 1$, $u_k = 2$ and $u_l = 3$. The indices do not have to be adjacent.

Count the replacements for which the number of climbs in the resulting string is divisible by 4. Two replacements are different if some character `?` is replaced by different digits in them. Since the answer can be large, print it modulo 998 244 353.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

Each of the next t lines contains a non-empty string s consisting of the characters `0`, `1`, `2`, `3` and `?`.

The total length of all strings does not exceed $5 \cdot 10^5$.

Output

For each test case print one integer — the number of replacements for which the number of climbs is divisible by 4, modulo 998 244 353.

Examples

Input	Output
6 0123 0?23 00112233 ???? 3210 ?0?1?2?3?	0 3 1 255 1 500

Note

In the first test case s contains no characters `?` and exactly one climb. The number 1 is not divisible by 4, so the answer is 0.

In the second test case s contains one character `?`. If it is replaced by 0, 2 or 3, the resulting string has no climbs, and 0 is divisible by 4. If it is replaced by 1, the resulting string is `0123`, which has one climb. So the answer is 3.

In the third test case every climb uses one of two occurrences of each digit, so there are $2 \cdot 2 \cdot 2 \cdot 2 = 16$ climbs, and 16 is divisible by 4. There are no characters `?`, so the answer is 1.

In the fourth test case s consists of four characters `?`, so there are $4^4 = 256$ replacements. Only the string `0123` has a climb (exactly one), and all the others have none, so the answer is 255.

In the fifth test case s contains no climbs.

G Grouped Vertices

Time limit 3 s · Memory limit 256 MiB · Standard input/output

You are given a tree with n vertices numbered from 1 to n. The *distance* between two vertices is the number of edges on the simple path between them.

A non-empty set S of vertices is called *connected* if for every two vertices of S all vertices of the simple path between them also belong to S.

For every x from 1 to n, find the maximum size of a connected set S such that the distance between any two vertices of S is at most x.

Input

The first line contains a single integer t ($1 \le t \le 5000$) — the number of test cases.

The first line of each test case contains an integer n ($1 \le n \le 5000$) — the number of vertices. Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n$, $u \ne v$) — an edge between vertices u and v. It is guaranteed that the edges form a tree.

The sum of n over all test cases does not exceed 5000.

Output

For each test case print one line with n integers: the x-th of them must be the maximum size of a connected set of vertices in which the distance between any two vertices is at most x.

Examples

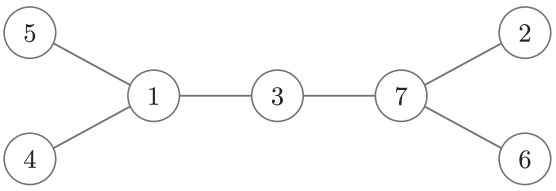
Input

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

Output

```
2 4 5 7 7 7 7
1
2 3 4 4
```

Note



In the first test case the tree has 7 vertices: vertex 1 is adjacent to 5, 3 and 4, and vertex 7 is adjacent to 3, 2 and 6.

- For $x = 1$ the answer is 2, for example $S = \{1, 3\}$.

- For $x = 2$ the answer is 4, for example $S = \{1, 3, 4, 5\}$.
- For $x = 3$ the answer is 5, for example $S = \{1, 3, 4, 5, 7\}$: the largest distances are between 5 and 7 and between 4 and 7, and they are equal to 3.
- For $x \geq 4$ the answer is 7, since the set of all vertices has largest distance 4.

In the second test case the tree consists of a single vertex, so the answer is 1.



In the third test case the tree is the chain $1 - 2 - 3 - 4$. A connected set of k vertices of a chain has largest distance $k - 1$, so the answers for $x = 1, 2, 3, 4$ are $2, 3, 4, 4$.

H Hall of Songs

Time limit 1 s · Memory limit 256 MiB · Standard input/output

A tavern has n musicians. At the beginning every musician knows exactly one song, and all n songs are different.

After each evening's performance the innkeeper picks a musician a uniformly at random among all n musicians, and then a musician b uniformly at random among the other $n-1$ musicians, so $b \neq a$. Musician a forgets the song they know and learns the song that musician b knows at that moment. All choices are independent. Thus every musician always knows exactly one song, although several musicians may know the same song.

Find the expected number of different songs known by the musicians after k evenings.

Input

The only line contains two integers n and k ($2 \le n \le 100$, $0 \le k \le 10^{18}$) — the number of musicians and the number of evenings.

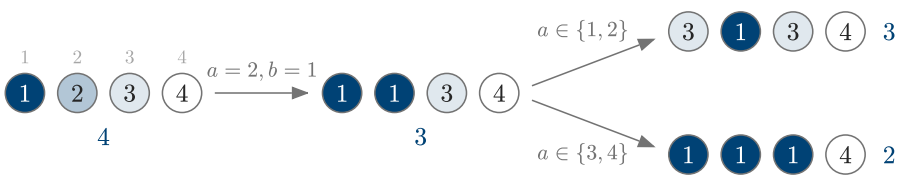
Output

Print one integer. It can be shown that the expected value is a rational number $\frac{p}{q}$, where q is not divisible by $998\,244\,353$. Print $p \cdot q^{-1} \bmod 998\,244\,353$, that is, the integer x such that $0 \le x < 998\,244\,353$ and $x \cdot q \equiv p \pmod{998\,244\,353}$.

Examples

Input	Output
4 2	499122179
6 1	5
2 1000000000000000000	1
5 0	5

Note



In the first example, $n = 4$ and $k = 2$. After the first evening two of the four musicians know the same song, and each of the other two knows a song of their own, so there are 3 songs. On the second evening, if a is one of the two musicians who know a song of their own, that song is lost and 2 songs remain; otherwise 3 songs remain. Each of these cases happens with probability $\frac{2}{4}$, so the expected number of songs is $\frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 3 = \frac{5}{2}$, and $499\,122\,179 \cdot 2 \equiv 5 \pmod{998\,244\,353}$. The figure shows one possible first evening ($a = 2$, $b = 1$; each circle shows the song of a musician, the musicians are num-

bered above) and both kinds of outcomes of the second evening; the numbers below and on the right are the numbers of different songs.

In the second example, $n = 6$ and $k = 1$. On the first evening musician a replaces their song by a different one, so exactly one song is lost and 5 songs remain.

In the third example, $n = 2$. After the first evening both musicians know the same song, and from then on exactly 1 song is known.

In the fourth example, $k = 0$, so all 5 songs are still known.

I Inconvenient Swaps

Time limit 1 s · Memory limit 256 MiB · Standard input/output

You are given an array $a_1, a_2, \ldots, a_n$ of integers.

In one operation you may choose two different indices i and j such that $a_i - a_j$ is divisible by 3 and swap the values a_i and a_j. The indices do not have to be adjacent. You may perform any number of operations, possibly zero.

Find the lexicographically smallest array that can be obtained.

An array $b_1, \ldots, b_n$ is *lexicographically smaller* than an array $c_1, \ldots, c_n$ if $b_i < c_i$ at the first position i where they differ. The elements are compared as integers, so, for example, $-5 < 2$.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

Each test case consists of two lines. The first line contains an integer n ($1 \le n \le 3 \cdot 10^5$) — the length of the array. The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$).

The sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case print one line with n integers: the lexicographically smallest array that can be obtained.

Examples

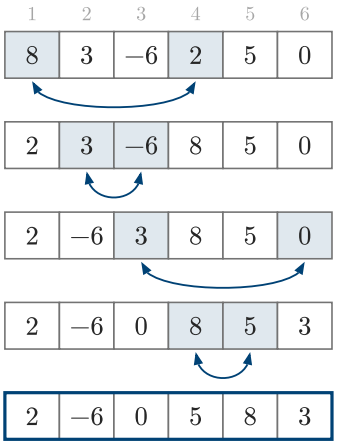
Input

```
3
6
8 3 -6 2 5 0
5
10 -1 7 -4 4
3
3 2 1
```

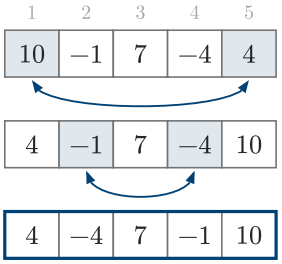
Output

```
2 -6 0 5 8 3
4 -4 7 -1 10
3 2 1
```

Note



In the first test case one can swap a_1 and a_4, then a_2 and a_3, then a_3 and a_6, and then a_4 and a_5. The array becomes $2, -6, 0, 5, 8, 3$. It can be shown that no smaller array can be obtained.



In the second test case one can swap a_1 and a_5, and then a_2 and a_4, which gives $4, -4, 7, -1, 10$.

In the third test case no difference of two elements is divisible by 3, so no operation is possible and the array does not change.

J Jar of Colours

Time limit 1.5 s · Memory limit 256 MiB · Standard input/output

Bohdana is repainting a fence made of n planks, numbered from 1 to n from left to right. Plank i currently has colour c_i. Colours are positive integers, and several planks may share a colour.

The paint on the fence is still wet, so Bohdana repaints it with *strokes*. In one stroke she chooses two planks l and r with $1 \le l \le r \le n$, picks up the current colour of plank l on a dry brush, and drags the brush to the right up to plank r. After the stroke, all planks $l, l+1, \dots, r$ have the colour that plank l had just before the stroke, and the other planks do not change.

Bohdana may make any number of strokes, possibly none, one after another. How many different colourings of the fence can she obtain? Two colourings are different if at least one plank has different colours in them. The initial colouring is counted too.

Since the answer can be very large, print it modulo 998 244 353.

Input

The first line contains a single integer t ($1 \le t \le 5000$) — the number of test cases.

Each test case consists of two lines. The first line contains an integer n ($1 \le n \le 5000$) — the number of planks. The second line contains n integers $c_1, c_2, \dots, c_n$ ($1 \le c_i \le 10^9$) — the colours of the planks from left to right.

The sum of n over all test cases does not exceed 5000.

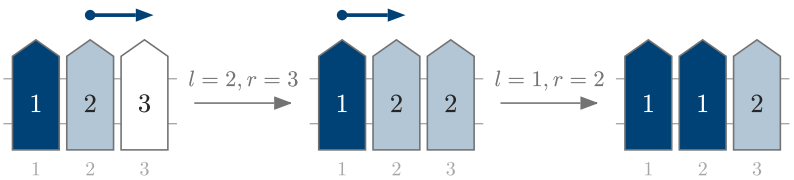
Output

For each test case print one integer — the number of different colourings Bohdana can obtain, modulo 998 244 353.

Examples

Input	Output
4 3 1 2 3 4 5 7 5 7 5 4 4 4 4 4 6 1000000000 3 3 1000000000 3 8	5 8 1 34

Note



In the first test case the fence is $(1, 2, 3)$, and the 5 possible colourings are $(1, 2, 3)$, $(1, 1, 3)$, $(1, 2, 2)$, $(1, 1, 2)$ and $(1, 1, 1)$. For example, $(1, 1, 2)$ is obtained by the stroke $l = 2$, $r = 3$, which gives

$(1, 2, 2)$, followed by the stroke $l = 1$, $r = 2$. The colouring $(1, 3, 2)$ cannot be obtained. The figure shows these two strokes: the arrow above the fence is the path of the brush.

In the second test case the 8 possible colourings are $(5, 7, 5, 7)$, $(5, 7, 5, 5)$, $(5, 7, 7, 5)$, $(5, 7, 7, 7)$, $(5, 5, 7, 5)$, $(5, 5, 7, 7)$, $(5, 5, 5, 7)$ and $(5, 5, 5, 5)$. For example, $(5, 7, 7, 5)$ is obtained by the stroke $l = 3$, $r = 4$ followed by the stroke $l = 2$, $r = 3$.

In the third test case all planks already have the same colour, and no stroke changes anything.

K Know This Problem?

Time limit 1 s · Memory limit 256 MiB · Standard input/output

You are given an array $h_1, h_2, \ldots, h_n$ of integers. Split it into two subsequences, that is, assign every element to exactly one of two subsequences (either of them may be empty), so that:

- the first subsequence is *non-decreasing*: if it consists of the elements with indices $i_1 < i_2 < \ldots < i_k$, then $h_{i_1} \le h_{i_2} \le \ldots \le h_{i_k}$;
- the second subsequence is *non-increasing*: if it consists of the elements with indices $j_1 < j_2 < \ldots < j_l$, then $h_{j_1} \ge h_{j_2} \ge \ldots \ge h_{j_l}$.

Determine whether such a split exists.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

Each test case consists of two lines. The first line contains an integer n ($1 \le n \le 10^6$) — the length of the array. The second line contains n integers $h_1, h_2, \ldots, h_n$ ($0 \le h_i \le 10^9$).

The sum of n over all test cases does not exceed 10^6.

Output

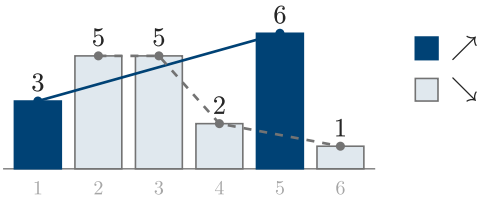
For each test case, print a line `YES` if a suitable split exists, and `NO` otherwise.

The words `YES` and `NO` may be printed in any letter case.

Examples

Input	Output
5 6 3 5 5 2 6 1 4 2 1 4 3 3 7 7 7 1 0 5 4 1 4 1 4	YES NO YES YES YES

Note



In the first test case $h = [3, 5, 5, 2, 6, 1]$, and a suitable split exists, so the answer is `YES`.

- For example, the non-decreasing subsequence may be $3, 6$, and the non-increasing subsequence $5, 5, 2, 1$.

- Another suitable split: the non-decreasing subsequence is $3, 5, 5, 6$, and the non-increasing subsequence is $2, 1$.

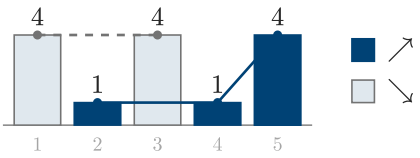
The figure shows the first of these splits: the non-decreasing subsequence is dark (solid line), and the non-increasing one is light (dashed line).

In the second test case $h = [2, 1, 4, 3]$. Suppose that the element 4 belongs to the non-increasing subsequence.

- Then the earlier and smaller elements 2 and 1 cannot belong to it, so both belong to the non-decreasing subsequence, but that subsequence cannot contain 2 followed by 1.
- So the element 4 belongs to the non-decreasing subsequence, and then the later and smaller element 3 belongs to the non-increasing one.
- Now the elements 2 and 1 are earlier and smaller than 3, so they cannot belong to the non-increasing subsequence, and again both would belong to the non-decreasing one, which is impossible.

Hence the answer is `NO`.

In the third test case all elements can belong to one subsequence, and the other one stays empty.



In the fifth test case the non-increasing subsequence is $4, 4$, and the non-decreasing subsequence is $1, 1, 4$.

L Lost List

Time limit 1 s · Memory limit 256 MiB · Standard input/output

A ski resort has slopes of n steepness levels, numbered from 1 to n (a higher number means a steeper slope), and m length levels, numbered from 1 to m (a higher number means a longer slope). A slope with steepness level i and length level j is said to have *type* (i, j).

Every skier has two limits: the maximum steepness level x and the maximum length level y that they agree to ride ($1 \le x \le n$, $1 \le y \le m$). A skier with limits (x, y) finds a slope of type (i, j) *suitable* if $i \le x$ and $j \le y$. Different skiers may have the same limits.

The resort ran a survey: for every slope type (i, j) it is known how many skiers find it suitable, and this number is $a_{i,j}$. The list of skiers is lost. For every pair of limits (x, y), find how many skiers have exactly these limits, or report that the survey is inconsistent, that is, no set of skiers leads to these numbers.

Input

The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases.

The first line of each test case contains two integers n and m ($1 \le n, m \le 1000$) — the number of steepness levels and the number of length levels.

The i-th of the next n lines contains m integers $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ ($0 \le a_{i,j} \le 10^9$) — the numbers of skiers who find the slopes with steepness level i and length levels $1, 2, \ldots, m$ suitable.

The sum of $n \cdot m$ over all test cases does not exceed 10^6.

Output

For each test case, if the survey is inconsistent, print a single line with the number -1.

Otherwise print n lines with m numbers each: the x-th line must contain $c_{x,1}, c_{x,2}, \ldots, c_{x,m}$, where $c_{x,y}$ is the number of skiers with limits (x, y).

Examples

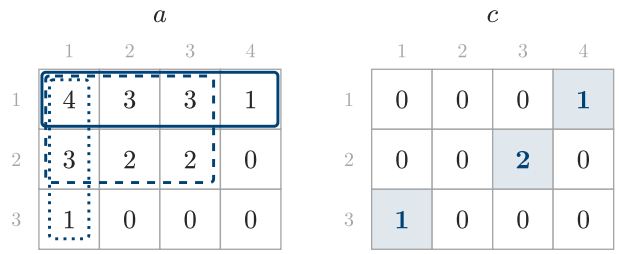
Input

```
3
3 4
4 3 3 1
3 2 2 0
1 0 0 0
2 2
2 1
1 1
2 2
1 2
1 1
```

Output

```
0 0 0 1
0 0 2 0
1 0 0 0
1 0
0 1
-1
```

Note



In the first test case there are four skiers: one with limits $(1, 4)$, two with limits $(2, 3)$ and one with limits $(3, 1)$. In the figure, the slopes suitable for each of these kinds of skiers are outlined on the left, and the answer is on the right. For example, the slope of type $(1, 1)$ is suitable for all of them, that is, for $1 + 2 + 1 = 4$ skiers, and the slope of type $(2, 2)$ is suitable only for the two skiers with limits $(2, 3)$.

M Mosaic Arch

Time limit 1 s · Memory limit 256 MiB · Standard input/output

You are given a grid with n rows and m columns. The rows are numbered from 1 to n from top to bottom, and the columns from 1 to m from left to right. Every cell (i, j), in row i and column j, has a colour (blue or yellow) and a weight $a_{i,j}$.

A Π-*shape* is defined by four integers r, c_1, c_2 and k such that $c_2 - c_1 \geq 2$, $k \geq 2$, $1 \leq c_1 < c_2 \leq m$ and $1 \leq r$, $r + k - 1 \leq n$. It consists of the cells, listed in this order:

$$(r + k - 1, c_1), \ldots, (r + 1, c_1), (r, c_1), (r, c_1 + 1), \ldots, (r, c_2), (r + 1, c_2), \ldots, (r + k - 1, c_2).$$

That is, the shape goes up along column c_1 from row $r + k - 1$ to row r, then right along row r from column c_1 to column c_2, and then down along column c_2 to row $r + k - 1$. The shape cannot be rotated or flipped.

A Π-shape is called *good* if any two consecutive cells in the list above have different colours. The weight of a Π-shape is the sum of $a_{i,j}$ over all its $2k + c_2 - c_1 - 1$ cells.

Find the largest weight of a good Π-shape in the grid, or report that there is none.

Input

The first line contains a single integer t ($1 \leq t \leq 2 \cdot 10^5$) — the number of test cases.

The first line of each test case contains two integers n and m ($1 \leq n, m$; $n \cdot m \leq 2 \cdot 10^5$) — the size of the grid.

Each of the next n lines contains a string of m characters describing a row of the grid from left to right. The character `B` stands for a blue cell, and `Y` for a yellow one.

Each of the next n lines contains m integers $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ ($1 \leq a_{i,j} \leq 10^9$) — the weights of the cells of row i.

The sum of $n \cdot m$ over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case print one integer: the largest weight of a good Π-shape, or -1 if there is no good Π-shape.

Examples

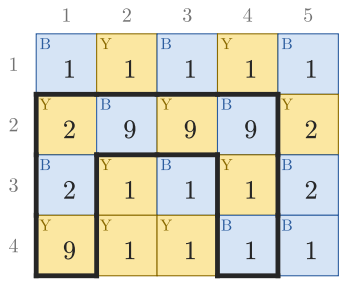
Input

```
4
4 5
BYBYB
YBYBY
BYBYB
YYYBB
1 1 1 1 1
2 9 9 9 2
2 1 1 1 2
9 1 1 1 1
2 3
YBY
BYB
1 2 3
4 5 6
3 4
YBYB
YYYY
BYBY
5 5 5 5
5 5 5 5
5 5 5 5
1 5
BYBYB
7 7 7 7 7
```

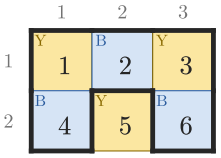
Output

```
42
16
25
-1
```

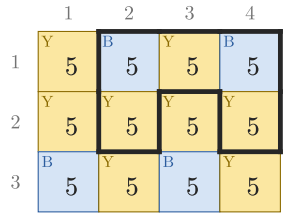
Note



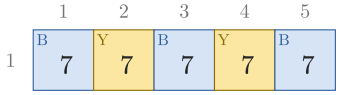
In the first test case the best Π-shape has $r = 2$, $c_1 = 1$, $c_2 = 4$ and $k = 3$. It consists of the cells $(4, 1), (3, 1), (2, 1), (2, 2), (2, 3), (2, 4), (3, 4), (4, 4)$, whose colours are yellow, blue, yellow, blue, yellow, blue, yellow, blue, so it is good. Its weight is $9 + 2 + 2 + 9 + 9 + 9 + 1 + 1 = 42$.



In the second test case the only good Π-shape consists of all cells except $(2, 2)$, and its weight is $1 + 2 + 3 + 4 + 6 = 16$.



In the third test case the only good Π-shape has $r = 1$, $c_1 = 2$, $c_2 = 4$ and $k = 2$.



In the fourth test case the grid has a single row, so $k \geq 2$ is impossible and there is no Π-shape at all.