A. Olympiad Date

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

The final of the first Olympiad by IT Campus "NEIMARK" is scheduled for March 1, 2025. A nameless intern was tasked with forming the date of the Olympiad using digits — 01.03.2025.

To accomplish this, the intern took a large bag of digits and began drawing them one by one. In total, he drew n digits — the digit a_i was drawn in the i-th turn.

You suspect that the intern did extra work. Determine at which step the intern could have first assembled the digits to form the date of the Olympiad (the separating dots can be ignored), or report that it is impossible to form this date from the drawn digits. Note that leading zeros **must be displayed**.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 20$).

The second line of each test case contains n integers a_i ($0 \le a_i \le 9$) — the numbers that the intern pulled out in chronological order.

Output

For each test case, output the minimum number of digits that the intern could pull out. If all the digits cannot be used to make a date, output the number 0.

Standard Input	Standard Output
4 10 2 0 1 2 3 2 5 0 0 1 8 2 0 1 2 3 2 5 0 8 2 0 1 0 3 2 5 0 16 2 3 1 2 3 0 1 9 2 1 0 3 5 4 0 3	9 0 8 15

,

B. Team Training

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

At the IT Campus "NEIMARK", there are training sessions in competitive programming — both individual and team-based!

For the next team training session, n students will attend, and the skill of the i-th student is given by a positive integer a_i.

The coach considers a team strong if its *strength* is at least x. The *strength* of a team is calculated as the number of team members multiplied by the minimum skill among the team members.

For example, if a team consists of 4 members with skills $[5, 3, 6, 8]$, then the team's *strength* is $4 \cdot min([5, 3, 6, 8]) = 12$.

Output the maximum possible number of strong teams, given that each team must have at least one participant and every participant must belong to exactly one team.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n and x ($1 \le n \le 2 \cdot 10^5, 1 \le x \le 10^9$) — the number of students in training and the minimum *strength* of a team to be considered strong.

The second line of each test case contains n integers a_i ($1 \le a_i \le 10^9$) — the skill of each student.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the maximum possible number of teams with *strength* at least x.

Standard Input	Standard Output
5 6 4 4 5 3 3 2 6 4 10 4 2 1 3 5 3 5 3 2 3 2 3 6 9 1 7 6 10 6 1 3 6 3 2	4 0 4 2 1

C. Combination Lock

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

At the IT Campus "NEIMARK", there are several top-secret rooms where problems for major programming competitions are developed. To enter one of these rooms, you must unlock a circular lock by selecting the correct code. This code is updated every day.

Today's code is a permutation* of the numbers from 1 to n, with the property that in every cyclic shift† of it, there is exactly one fixed point. That is, in every cyclic shift, there exists exactly one element whose value is equal to its position in the permutation.

Output any valid permutation that satisfies this condition. Keep in mind that a valid permutation might not exist, then output -1.

---

*A permutation is defined as a sequence of length n consisting of integers from 1 to n, where each number appears exactly once. For example, (2 1 3), (1), (4 3 1 2) are permutations; (1 2 2), (3), (1 3 2 5) are not.

†A cyclic shift of an array is obtained by moving the last element to the beginning of the array. A permutation of length n has exactly n cyclic shifts.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 500$). The description of the test cases follows.

A single line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the desired permutation. If multiple solutions exist, output any one of them. If no suitable permutations exist, output -1.

Standard Input	Standard Output
3 4 5 3	-1 4 1 3 5 2 1 3 2

Note

In the second example, there is a permutation such that in each cyclic shift there is a fixed point (highlighted in dark red):

1	2	3	4	5
4	1	3	5	2

1	2	3	4	5
2	4	1	3	5

1	2	3	4	5
5	2	4	1	3

1	2	3	4	5
3	5	2	4	1

1	2	3	4	5
1	3	5	2	4

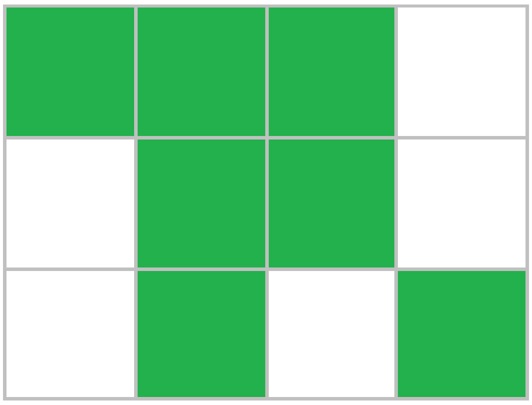
The first line contains the element numbers, and the second line contains all the shifts of the desired permutation.

D. Place of the Olympiad

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

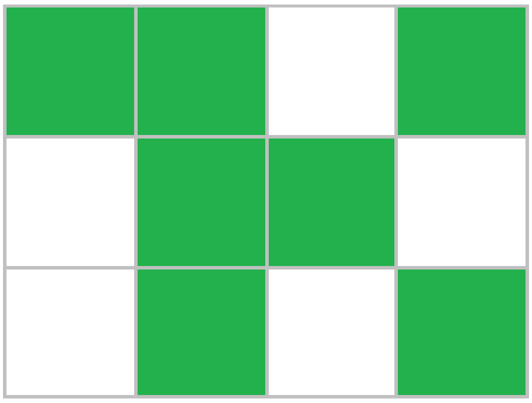
For the final of the first Olympiad by IT Campus "NEIMARK", a rectangular venue was prepared. You may assume that the venue is divided into n rows, each containing m spots for participants' desks. A total of k participants have registered for the final, and each participant will sit at an individual desk. Now, the organizing committee must choose the locations for the desks in the venue.

Each desk occupies one of the m spots in a row. Moreover, if several desks occupy consecutive spots in the same row, we call such a group of desks a *bench*, and the number of desks in the group is the bench's length. For example, seating 7 participants on a 3×4 venue (with $n = 3, m = 4$) can be arranged as follows:



In the figure above, the first row has one bench of length 3, the second row has one bench of length 2, and the third row has two benches of length 1.

The organizing committee wants to choose the locations so that the length of the longest bench is as small as possible. In particular, the same 7 desks can be arranged in a more optimal way, so that the lengths of all benches do not exceed 2:



Given the integers n, m, and k, determine the minimum possible length of the longest bench.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

A single line of each test case contains three positive integers — n, m, k ($1 \le n, m, k \le 10^9$, $k \le n \cdot m$).

Output

For each test case, output a single number — the minimum possible length of the longest bench.

Standard Input	Standard Output
5 3 4 7 5 5 5 1 13 2 2 4 7 1 5 4	2 1 1 4 2

,

E. Interesting Ratio

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

Recently, Misha at the IT Campus "NEIMARK" camp learned a new topic — the Euclidean algorithm.

He was somewhat surprised when he realized that $a \cdot b = lcm(a, b) \cdot gcd(a, b)$, where $gcd(a, b)$ — is the greatest common divisor (GCD) of the numbers a and b and $lcm(a, b)$ — is the least common multiple (LCM). Misha thought that since the product of LCM and GCD exists, it might be interesting to consider their quotient: $F(a, b) = \frac{lcm(a,b)}{gcd(a,b)}$.

For example, he took $a = 2$ and $b = 4$, computed $F(2, 4) = \frac{4}{2} = 2$ and obtained a prime number (a number is prime if it has exactly two divisors)! Now he considers $F(a, b)$ to be an interesting ratio if $a < b$ and $F(a, b)$ is a prime number.

Since Misha has just started studying number theory, he needs your help to calculate — how many different pairs of numbers a and b are there such that $F(a, b)$ is an interesting ratio and $1 \le a < b \le n$?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^3$). The description of the test cases follows.

A single line of each test case contains a single integer n ($2 \le n \le 10^7$).

It is guaranteed that the sum of n over all test cases does not exceed 10^7.

Output

For each test case, output the number of interesting ratios $F(a, b)$ for pairs satisfying $1 \le a < b \le n$.

Standard Input	Standard Output
4 5 10 34 10007	4 11 49 24317

F. Igor and Mountain

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

The visitors of the IT Campus "NEIMARK" are not only strong programmers but also physically robust individuals! Some practice swimming, some rowing, and some rock climbing!

Master Igor is a prominent figure in the local rock climbing community. One day, he went on a mountain hike to ascend one of the peaks. As an experienced climber, Igor decided not to follow the established trails but to use his skills to climb strictly vertically.

Igor found a rectangular vertical section of the mountain and mentally divided it into n horizontal levels. He then split each level into m segments using vertical partitions. Upon inspecting these segments, Igor discovered convenient protrusions that can be grasped (hereafter referred to as *holds*). Thus, the selected part of the mountain can be represented as an $n \times m$ rectangle, with some cells containing holds.

Being an experienced programmer, Igor decided to count the number of valid *routes*. A route is defined as a sequence of **distinct** holds. A route is considered valid if the following conditions are satisfied:

- The first hold in the route is located on the very bottom level (row n);
- The last hold in the route is located on the very top level (row 1);
- Each subsequent hold is not lower than the previous one;
- At least one hold is used on each level (i.e., in every row of the rectangle);
- At most two holds are used on each level (since Igor has only two hands);
- Igor can reach from the current hold to the next one if the distance between the centers of the corresponding sections does not exceed Igor's arm span.

Igor's arm span is d, which means he can move from one hold to another if the **Euclidean distance** between the centers of the corresponding segments does not exceed d. The distance between sections (i_1, j_1) and (i_2, j_2) is given by $\sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}$.

Calculate the number of different valid routes. Two routes are considered different if they differ in the list of holds used or in the order in which these holds are visited.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^3$). The description of the test cases follows.

The first line of each test case contains three integers n, m, d ($2 \le n \le 2000$, $1 \le m, d \le 2000$).

Each of the following n lines contains m characters — the description of the corresponding level of the mountain. The symbol '#' represents an empty section, and the symbol 'X' represents a section with a hold. The levels are described from top to bottom.

It is guaranteed that the sum of $n \cdot m$ over all test cases does not exceed $4 \cdot 10^6$.

Output

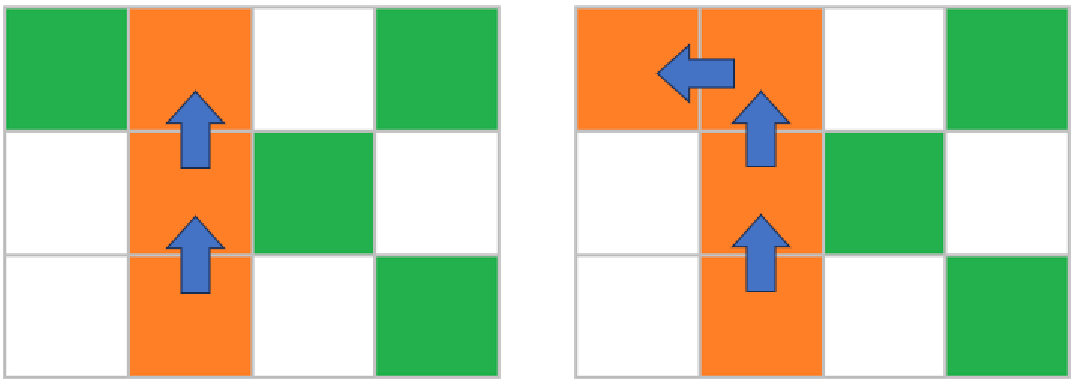
For each test case, output the number of different routes modulo 998244353.

Standard Input	Standard Output

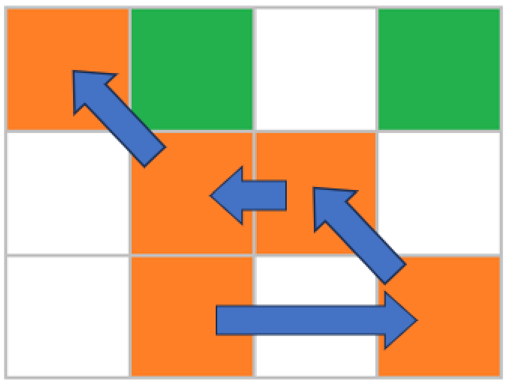
3 3 4 1 XX#X #XX# #X#X 3 4 2 XX#X #XX# #X#X 3 1 3 X X #	2 60 0

Note

Possible routes in the first case:



In the second example, Igor's arm span has become larger, so new routes are available to him, for example this one:



In the third example, there are no holds on the lower level, so there are no correct routes.

G. Gleb and Boating

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

Programmer Gleb frequently visits the IT Campus "NEIMARK" to participate in programming training sessions.

Not only is Gleb a programmer, but he is also a renowned rower, so he covers part of his journey from home to the campus by kayaking along a river. Assume that Gleb starts at point 0 and must reach point s (i.e., travel s meters along a straight line). To make the challenge tougher, Gleb has decided not to go outside the segment $[0, s]$. The dimensions of the kayak can be neglected.

Gleb is a strong programmer! Initially, his power is k. Gleb's power directly affects the movement of his kayak. If his current power is x, then with one paddle stroke the kayak moves x meters in the current direction. Gleb can turn around and continue moving in the opposite direction, but such a maneuver is quite challenging, and after each turn, his power decreases by 1. The power can never become 0 — if his current power is 1, then even after turning it remains 1. Moreover, Gleb cannot make two turns in a row — after each turn, he must move at least once before making another turn. Similarly, Gleb cannot make a turn immediately after the start — he must first perform a paddle stroke.

Gleb wants to reach point s from point 0 without leaving the segment $[0, s]$ and while preserving as much power as possible. Help him — given the value s and his initial power k, determine the maximum possible power he can have upon reaching point s.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

A single line of each test case contains two integers s and k ($1 \le s \le 10^9$, $1 \le k \le 1000$, $k \le s$).

It is guaranteed that the sum of k over all test cases does not exceed 2000.

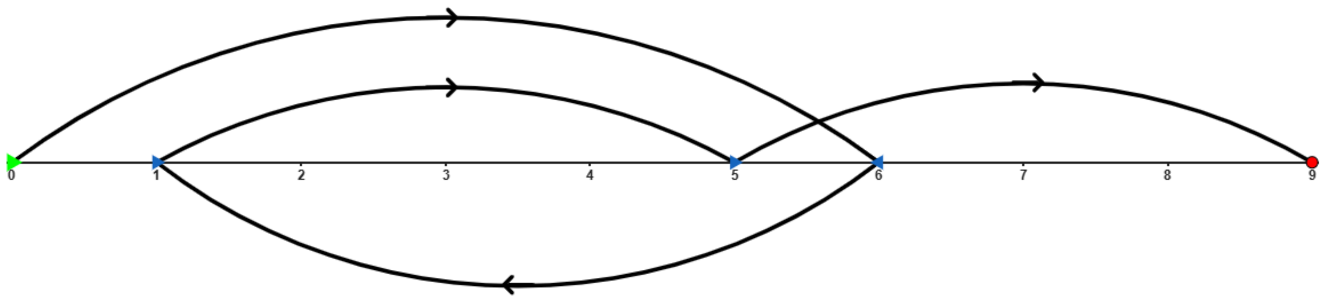
Output

For each test case, output the maximum possible power Gleb can have at the end of his journey.

Standard Input	Standard Output
8 9 6 10 7 24 2 123456 777 6 4 99 6 10 4 99 4	4 1 2 775 1 4 2 2

Note

One of the variants of Gleb's movement in the first example:


0
1
2
3
4
5
6
7
8
9