

Problem A

The n Days of Christmas

Time limit: 1 second

This is a **SCORED** problem. There are ten subgroups of data. The subgroups will be judged independently. Solving all of the data in a subgroup will score 0.1 of a problem (That's how it will look on the scoreboard, but it may say 10 in Kattis' response). Solving all of the data in all ten subgroups will score 1 full problem. You may continue to make submissions to try to improve your score.

Time penalty for this problem will be assessed as follows:

- One point for every minute from the start of the contest to the last submission that increased your score.
- Twenty points for every submission before that last score-increasing submission that did not increase your score, with the exception of compiler errors, which are not assessed penalties.
- No points are assessed if your score never rises above zero

Remember the song "The 12 Days of Christmas?"

"On the first day of Christmas my true love gave to me, a partridge in a pear tree."

Great. I wanted an iPhone.

"On the second day of Christmas my true love gave to me, two turtle doves, and a partridge in a pear tree."

OK, now I have four birds: two turtle doves and a partridge today, and a partridge from yesterday. My cats are going crazy.

"On the third day of Christmas my true love gave to me, three French hens, two turtle doves, and a partridge in a pear tree."

What's with the birds? That's six more today, so I now have ten of them, and no place to put them (except inside the cats). How long can my true love keep this up?

Given an integer n , how many gifts have I received after n days? Since this value may be large, output it modulo 998,244,353.



Input

The single line of input contains an integer n , which is the number of days my true love gives me gifts.

There are ten subgroups of data, each with different limits on n :

1. $(1 \leq n \leq 10)$
2. $(10 < n \leq 1,000)$
3. $(1,000 < n \leq 10^5)$
4. $(10^5 < n \leq 10^7)$
5. $(10^7 < n \leq 10^9)$
6. $(10^9 < n \leq 10^{11})$
7. $(10^{11} < n \leq 10^{13})$
8. $(10^{13} < n \leq 10^{15})$
9. $(10^{15} < n \leq 10^{17})$
10. $(10^{17} < n \leq 10^{18})$

Output

Output a single integer, which is the number of gifts I receive after exactly n days. Because this number may be very large, output it modulo 998,244,353.

Sample Input 1

1	1
---	---

Sample Output 1**Sample Input 2**

2	4
---	---

Sample Output 2**Sample Input 3**

3	10
---	----

Sample Output 3

Problem B

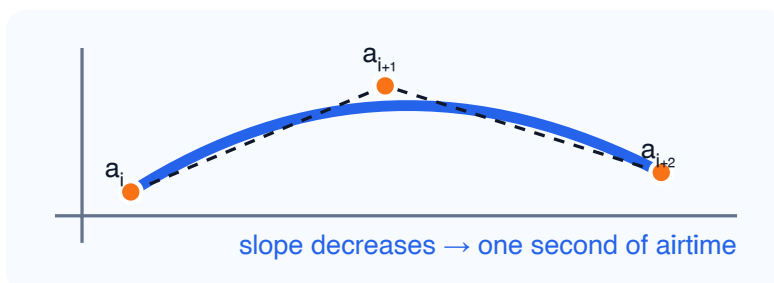
Airtime

Time limit: 2 seconds

Fred loves roller coasters! He particularly loves airtime – that’s where the coaster goes over an element that causes riders to float out of their seats!

Fred has been doing a lot of internet research, and he’s found a lot of data on a lot of roller coasters. He has collected many altitude profiles, which give the height of the track at many points along the course, in order. These measurements have been taken from one of each coaster’s trains at regular one-second intervals.

Fred knows that he’ll get some airtime if the track forms an arcing shape. Formally, given three consecutive altitude measurements a_i , a_{i+1} and a_{i+2} at times t_i , t_{i+1} and t_{i+2} , he’ll get a second of airtime iff the slope from (t_i, a_i) to (t_{i+1}, a_{i+1}) is strictly greater than the slope from (t_{i+1}, a_{i+1}) to (t_{i+2}, a_{i+2})



Given the altitude profile of a roller coaster, determine how many seconds of airtime Fred can experience.

Input

The first line of input contains an integer n ($3 \leq n \leq 10^5$), which is the number of altitude measurements to follow.

Each of the next n lines contains an integer a ($0 \leq a \leq 10^9$). This is the sequence of altitude measurements, in inches. They are given in order, one per second.

Output

Output a single integer, which is the total number of seconds of airtime Fred can experience on this roller coaster.



Sample Input 1

```
13
1
2
3
3
4
5
6
5
4
4
4
3
2
```

Sample Output 1

```
3
```

Sample Input 2

```
7
1
3
8
15
25
40
60
```

Sample Output 2

```
0
```

Problem C

Babylon

Time limit: 9 seconds

Abor and Tynna are building a tower again – the one they call, with a kind of doomed affection, the Tower of Babel. They have raised it before and watched it come down; each time, they clear the ground and begin from the very first block. Tonight they mean to build it all the way back up.

They have n blocks of pairwise distinct colors. Each block is a rectangle of some length and width that lies flat, represented as a footprint in the xy -plane; the tower itself rises upward, away from that plane, one level per block. Abor and Tynna intend to stack all n blocks into a single tower exactly n blocks tall – one block to each level, stacked directly one atop another. To build it, they have a stacking robot that can pick up a block, rotate it either 0 or 90 degrees (swapping its length and width), and place it so that the bottom-left corner of its footprint sits exactly at lattice point (x, y) .

For calibration, the first block the robot places must go at $(0, 0)$. For stability, no block may have any empty space strictly underneath it: the footprint of every block after the first must lie entirely within the footprint of the single block directly beneath it – that is, viewed from above, each block sits completely within the outline of the block below.

Two ways of constructing the tower are considered different if the order in which the blocks are placed differs, or if some block occupies a different footprint – that is, if its footprint covers a different set of points in the plane.

Compute the number of ways the tower can be constructed.

Input

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

The descriptions of t test cases follow. Each test case starts with a line containing a positive integer n ($1 \leq n \leq 10^5$), which is the number of blocks Abor and Tynna have.

Each of the next n lines contains two integers l and w ($1 \leq l, w \leq 10^9$), the length and width of a block.

Over all test cases, the sum of n is at most 10^5 .

Output

For each test case, output a single integer, which is the number of ways the tower can be constructed, one per line. Since this number may be large, output it modulo 998,244,353.



Sample Input 1

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

Sample Output 1

```
4
0
12
```

Problem D

Brownie Cutting

Time limit: 3 seconds

Alice has recently made a batch of brownies in an oddly shaped pan, represented as a convex polygon of n points. Alice has heard that people prefer brownie edges and in a misguided attempt to adhere to this principle, she has decided to focus on increasing the total perimeter of her brownies after they have been cut.

Specifically, she requires the sum of perimeters of all brownies after cutting to be at least k units greater than the original perimeter of the initial brownie. Alice is really, really into baking and has recently purchased a rather fancy and expensive brownie-cutting laser that can fire an infinitely thin, straight beam fully across the pan. In other words, each laser shot cuts along the intersection of an infinite straight line with the original polygon. After firing the laser zero or more times, when she is all finished, Alice will remove all the newly cut brownies from the pan. This is the only time the brownies move.

Alice will accept any brownie with a non-zero area (even if it is really tiny). For warranty reasons, the laser cannot be fired along the same path twice; that is, two laser paths cannot be coincident, although they can intersect. As the laser is very fancy and infinitely thin, it does not destroy or displace any brownie material when firing.

Since firing this beam is rather expensive, Alice would like to minimize the number of times the laser is used. Given the shape of Alice's initial brownie pan, determine the minimum number of times she must fire the laser so that the total perimeter of all resulting brownie pieces is increased by at least k .

Input

The first line of input contains two integers n ($3 \leq n \leq 2 \cdot 10^5$) and k ($0 \leq k \leq 10^9$), where n is the number of points that describe the original uncut brownie, and k is the minimum the perimeter must increase after all cuts are made.

Each of the next n lines contains two integers x and y ($0 \leq x, y \leq 10^9$) representing the points defining the original uncut brownie. The points will be given in counterclockwise order, and no three points will be collinear. The resulting polygon will be convex.

Output

Output a single integer, which is the minimum number of times Alice must fire the brownie cutting laser.



Sample Input 1

```
4 42
1 0
1 1
0 1
0 0
```

Sample Output 1

```
15
```

Problem E

Espresso Macchiato

Time limit: 1 second

Tommy likes two things: city planning and espresso macchiatos. His days are packed, he works around the clock, and he has barely a minute to stop and talk – but a man has his priorities, and his coffee is very important to him. So, when he is handed the job of planning a whole new city, he sees his chance: to guarantee himself exactly one macchiato on the way to work, every single morning, no stress.

The city is an $n \times n$ grid. Tommy's home sits in the top-left corner and his workplace in the bottom-right. He is forever in a rush, so on the walk from home to work he only ever moves right and down.

He wants to design the city so that both of the following hold:

- Each square in the grid has either no coffee shop or exactly one coffee shop. Tommy's home and workplace cannot have coffee shops.
- No matter how Tommy walks in the grid, his path takes him through exactly one square that has a coffee shop.

One macchiato every morning – never none, because he needs it, and never two, because who has the time? Help him design the city to fit his constraints!

Input

The first line of input contains a single integer t ($1 \leq t \leq 49$), which is the number of test cases.

The descriptions of t test cases follow. Each test case consists of a single line with a single integer n ($2 \leq n \leq 50$) which is the size of the city Tommy is planning.

Output

For each test case, output an $n \times n$ grid. Each character is either a `.` to indicate a square without a coffee shop or a `C` to indicate a square with a coffee shop.

If there are multiple city plans that are valid, any will be accepted. It can be proven that it is always possible to construct a valid city plan.



Sample Input 1

2
6
7

Sample Output 1

.....C
.....C.
...C..
..C...
.C....
C.....
.....
.....
.....
.....
.....C
.....C.
.....C..

Problem F

Fairytale

Time limit: 1 second

Alexander is writing his fairytale – the one he can’t quite let go of, about the girl he loved. He means to carry the whole of it as music on his violin, the way only he can, having composed it, sung it, and bowed it all himself.

A violin has four strings, and for Alexander each string is capable of producing 10 unique pitches. Specifically, the first string can produce pitches 1 through 10, the second pitches 11 through 20, the third 21 through 30, and the fourth 31 through 40.

He has finished the violin accompaniment, and before he plays it through he wants to know how far across the instrument the melody actually roams – how many of the four strings his fairytale calls for. Help him!

Input

The first line of input contains a single integer t ($1 \leq t \leq 50$), which is the number of test cases.

The descriptions of t test cases follow. Each test case starts with a line containing a single integer n ($1 \leq n \leq 50$), which is the number of notes in the accompaniment.

The next line contains n integers p ($1 \leq p \leq 40$), the pitches of the n notes in the accompaniment.

Output

For each test case, output a single integer, which is the number of strings that will be used, one per line.

Sample Input 1

```
4
3
1 5 10
4
11 20 31 40
5
7 10 11 20 29
6
1 11 21 30 31 40
```

Sample Output 1

```
1
2
3
4
```

This page is intentionally left blank.

Problem G

Last Dance

Time limit: 8 seconds

Stefania is running a dance night, and tonight she is the DJ. She plays one song after another. Each one is of some genre, and every time a track drops, part of the crowd gets up to dance to it. The night is one last dance after another: for anyone on the floor, only the newest dance counts – the moment they join a fresh song, that song becomes their last dance, and whatever they were dancing to before fades into the night behind it. Every so often Stefania leans into the mic and asks – for a given genre, how many people’s last dance was to a song of that genre? – and tallies what comes back, though she wants to know the true answers herself.

Help Stefania figure out how many people’s last dance was to a song of a given genre.

Input

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

The descriptions of t test cases follow. Each test case starts with a line containing three integers n ($1 \leq n \leq 10^9$), q ($1 \leq q \leq 10^6$), and g ($1 \leq g \leq 10^9$), where n is the number of people attending the event, q is the number of songs Stefania plays plus the number of questions Stefania asks, and g is the number of different genres of songs Stefania will play.

Each of the next q lines will take one of two forms:

- P s a i_1 $\dots$ i_a

Stefania plays a song with genre s ($1 \leq s \leq g$), and a ($1 \leq a \leq n$) people i_1 through i_a ($1 \leq i_x \leq n$) dance to it. All a people are distinct, and genre s becomes the most recent genre each of them has danced to, replacing whatever genre they had danced to before.

- Q s

Stefania wants to know how many people’s most recent dance was to a song of genre s ; that is, for how many people is s the genre of the last song they danced to.

Over all test cases, the sum of q is at most 10^6 , and the sum of a is at most 10^6 .

Output

For each query of the form Q s , output a single integer, which is the number of people whose most recent dance was to a song of genre s (the last song they danced to has genre s), one per line. A person who has not yet danced to any song is not counted for any genre.

**Sample Input 1**

```
1
5 5 3
P 1 3 1 2 3
Q 1
Q 2
P 2 2 2 3
Q 1
```

Sample Output 1

```
3
0
1
```

Sample Input 2

```
2
5 3 2
P 1 2 1 2
Q 1
Q 2
5 4 2
P 2 1 3
Q 1
Q 2
Q 2
```

Sample Output 2

```
2
0
0
1
1
```

Problem H

My System

Time limit: 3 seconds

Felicia has a system. It is how she keeps her head straight when everything wants to pull her back and forth – done with it one moment, right back in it the next, a loop she knows by heart. Her system holds most of the time. It was never meant to be perfect; it only has to be good enough. Now she wants to know whether you can find a system of your own.

She is giving you a true-false test. Here is what you know about the test.

- There are 200 questions on the test numbered 1 through 200.
- For each i from 1 to 100, questions $2i - 1$ and $2i$ have different answers. So, questions 1 and 2 will have different answers, as will questions 3 and 4, but there is no specified relationship between questions 2 and 3

You will answer questions one by one and Felicia will tell you whether you got the answers right or wrong.

Your goal is to get at least a 60% on her test. You do not need to be perfect – her system never was. Can you do it?

Interaction

This is an interactive problem.

Your program will be tested up to 100 times. For each test, interaction starts by reading a single integer t ($1 \leq t \leq 100$), the number of cases in that test.

The descriptions of t test cases follow. For each test case, you should repeat the following process 200 times:

- Output a single letter, either F or T. This represents your answer to the given question.
- Read in a single letter, either F or T. This represents the correct answer to the given question.

Do not forget to flush the output after each letter you output.

The interactor is adaptive – it cannot use your current guess to determine the correct answer for the current question, but it can use all previous guesses, including those in previous cases and tests. The correct answers will always respect the constraints as specified above.



If the interactor receives any invalid or unexpected input, the interactor will output -1 and then immediately terminate. Your program should cleanly exit in order to receive a Wrong Answer verdict, otherwise the verdict that it receives may be arbitrary.

If all of your responses are valid and you answer fewer than 60% of the questions correctly in a single test case, the interactor will also stop. The interactor will only stop on completion of the test, even if your score is guaranteed to be at least 60% or below 60% no matter how your remaining answers go. Cleanly exit as specified above.

Sample Interaction Explanation

The following table shows one possible beginning of an interaction. The interactor first reports that there are 100 test cases. The contestant then answers the first question of the first test case correctly. The interaction continues until all questions have been answered or the interactor stops it as described above.

Read	Sample Interaction 1	Write
100		
	F	
F		

Problem I

Only One Order

Time limit: 3 seconds

Anissa is in charge of judging a figure skating competition. In this competition, figure skaters must perform a routine of n unique elements numbered from 1 to n . Anissa's required routine is specifically to perform element 1 first, then element 2, and so on, ending with element n .

Unfortunately for Anissa, the skaters in this competition are all AIs, and instead of being told this required order directly, they have been given the following instructions instead:

- You are given m constraints, each consisting of an ordered pair (a, b) , indicating that element a must be performed before element b at some point.
- You must come up with a routine that consists of performing each of the n elements exactly once, in any order that satisfies all the constraints.

Anissa wants these instructions to allow exactly one routine: $1, 2, \dots, n$, in that order. She will add as few constraints as possible to ensure that all the skaters perform the routine exactly as requested. Help Anissa compute the minimum number of constraints she needs to add!

Input

The first line contains an integer t ($1 \leq t \leq 10^5$), which is the number of test cases.

The descriptions of t test cases follow. Each test case starts with a line containing two integers n and m ($1 \leq n \leq 10^5, 0 \leq m \leq 10^5$), where n is the number of elements, and m is the number of constraints.

Each of the next m lines contains two integers a and b ($1 \leq a, b \leq n, a \neq b$), indicating that element a must be performed before element b . All constraints are distinct.

Over all test cases, the sum of n is at most 10^5 and the sum of m is at most 10^5 .

Output

For each test case, output a single integer, which is the minimum number of constraints Anissa needs to add, one per line. If it is impossible for Anissa to make this happen, output -1 .



Sample Input 1

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

Sample Output 1

```
1
3
-1
```

Problem J

Regarde!

Time limit: 1 second

Monroe is in charge of Busan Library. Busan Library is known for its prestige but also its especially quiet environment.

One day, a group of rowdy computer programmers comes in and starts disturbing the peace, and Monroe has to step in to hush them. Depending on how severely the group is disturbing the peace, Monroe will hush with a specific intensity. Specifically, a hush of intensity n will cause Monroe to say `shh` . . . with $n + 1$ `h`'s.

Given the desired intensity of Monroe's hush, determine exactly how Monroe will hush the programmers.

Input

The first line of input contains a single integer t ($1 \leq t \leq 50$), which is the number of test cases.

The descriptions of t test cases follow. Each test case consists of a single line containing a single integer n ($1 \leq n \leq 50$), which is the intensity of Monroe's hush.

Output

For each test case, output the exact hush that Monroe will make, one per line.

Sample Input 1

```
2
1
10
```

Sample Output 1

```
shh
shhhhhhhhhhhh
```

This page is intentionally left blank.

Problem K

Six Seven

Time limit: 2 seconds

Audrey's favorite number is 67. She's in denial about the actual reason, but it's really because the digits, when read from left to right, increase by exactly one from digit to digit. As it turns out, Audrey happens to like exactly the numbers with this property.

One day, Audrey starts counting from a given integer going up until she reaches an integer she likes, at which point she loses all her focus and stops counting. Determine, for the starting integer, what integer she ends her counting at.

Please note that Audrey doesn't like 90 because 0 is not one more than 9 and Audrey doesn't like 910 because 10 is not a digit.

Input

The first line contains an integer t ($1 \leq t \leq 10^5$), which is the number of test cases.

The descriptions of t test cases follow. Each test case consists of a single line containing a single integer n ($1 \leq n \leq 10^8$), which is the integer where Audrey starts.

Output

For each test case, output an integer, which is the integer where Audrey stops, one per line. If Audrey would count forever, output -1 .

Sample Input 1

```
4
6
7
67
6767
```

Sample Output 1

```
6
7
67
6789
```

This page is intentionally left blank.

Problem L

Stardust Curry

Time limit: 2 seconds

Rajju Baba likes doing pushups! In particular, he most enjoys doing pushups while thinking about solving programming contest problems!

But Rajju Baba does not simply do pushups – he sinks into them, each set its own little trance, the reps dropping into a rhythm like a beat only he can hear. Push too hard, too fast, and the spell breaks; what he is truly after is stillness, the very lowest stress he can bring himself to feel.

During his next programming contest, Rajju Baba has set up n breaks in which he can do pushups. Each break i lasts for t_i seconds, and over the course of all n breaks he wants to do a total of exactly p pushups. He need not do a whole number of pushups in any given break – he may split his effort across the breaks however he likes, so the amount he does in a single break may be any nonnegative real number. If, during a break of s seconds, Rajju Baba does x pushups, then he will feel $\frac{x^2}{s}$ units of stress.

Help Rajju Baba minimize the sum of units of stress he feels over all his breaks, so he can slip back to the problems in peace.

Input

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

The descriptions of t test cases follow. Each test case starts with a line containing two integers n and p ($1 \leq n \leq 10^5, 1 \leq p \leq 10^9$), where n is the number of Rajju's breaks, and p is the number of pushups Rajju will do.

The next line contains n integers t_i ($1 \leq t_i \leq 10^9$), which are the lengths of the n breaks Rajju takes.

Over all test cases, the sum of n will not exceed 10^5 .

Output

For each test case, output a single real number, which is the minimum number of units of stress Rajju Baba will feel, one per line. Each answer will be considered correct if it has an absolute or relative error of at most 10^{-6} .



Sample Input 1

```
2
4 100
1 2 3 4
3 1
1 1 1
```

Sample Output 1

```
1000.0000000000
0.3333333333
```

Problem M

Tourism Balance

Time limit: 3 seconds

In the Boiling Isles, there are n cities, each of which has a certain value v_i . There are also n roads that connect the cities to one another. The roads connect the cities so that it is possible to travel from any city to any other city.

The government has realized that the valuable cities attract huge crowds, so they want to even out tourism across all cities. If you are traveling from city i to city j where $v_i < v_j$, the government will charge you a toll of $v_j - v_i$ dollars to discourage this kind of travel. On the other hand, if you are traveling from city i to city j where $v_i > v_j$, the government will pay you $v_i - v_j$ dollars to bolster tourism in city j .

As a tourist yourself, you want to follow a simple tourism path: an ordered sequence of cities connected by roads, with no city repeated. A path may consist of a single city; such a path contains no roads, and has a toll of 0. Paths with opposite directions, or with different intermediate cities, are different.

The net profit of a path is the money the government pays you minus the tolls you pay. You want this profit to be exactly the k dollars you spent on airplane tickets. How many simple tourism paths have a net profit of exactly k ?

Input

The first line contains two integers n and k ($3 \leq n \leq 10^5, 0 \leq k \leq 10^9$), where n is the number of cities and k is the profit you wish to make.

The next line has n space separated integers v_i ($1 \leq v_i \leq 10^9$), which are the values of the cities.

Each of the next n lines contain two integers a and b ($1 \leq a, b \leq n, a \neq b$), describing an undirected road that connects cities a and b . No road connects a city to itself, and no pair of cities is connected by more than one road.

Output

Output a single integer, which is the number of simple tourism paths that achieve a profit of exactly k dollars.

**Sample Input 1**

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

Sample Output 1

```
6
```

Sample Input 2

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

Sample Output 2

```
6
```

Sample Input 3

```
10 1
1 1 1 1 1 2 2 2 2 2
1 5
7 9
7 8
2 3
7 10
1 2
1 3
4 6
1 4
1 7
```

Sample Output 3

```
35
```