Problem A
Advance to Taoyuan Regional

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

In the last cycle, the domestic team selection process of ICPC Taoyuan Regional Contest chooses teams from preliminary contests according to the following categories.

- Category 1: Among the teams that participated in the "ICPC Taiwan Online Programming Contest" (TOPC) and solved one problem, the best team from each university will be selected. This category is limited to 30 teams.
- Category 2: The top 40 teams in the "National Collegiate Programming Contest" will be selected.
- Category 3: "Taiwan Private University Programming Contest" qualifies 10 best teams, but no more than two teams per university.
- Category 4: "Taiwan Technology University Programming Contest" qualifies 10 best teams, but no more than two teams per university.
- Category 5: 10 teams will be qualified for their CPE performance, but no more than one team from each university. Such teams must participate in TOPC and solve at least one problem.
- Category 6: If a team of the above five categories resigns for any reason and there are other teams from the same university participating in TOPC. The replacement of the resigning team will be selected from the TOPC participants from the same university according to their performance.
- Category 7: If the total number of teams in the above six categories is less than 100, the organizer will invite the top teams of TOPC to participate in the regional contest. The invited teams must be in the top half of the final standings.
- Category 8: Free to participate, priority will be given to universities not selected in the above categories and will be ranked according to the order of registration.

The domestic team selection rule of ICPC Taoyuan Regional Contest has a major change in the 2023-2024 cycle. The organizer added a note, "If the list of selected teams cannot be submitted 35 days prior to the ICPC Taoyuan Regional Contest, the category will be subject to suspension for that particular year." This change was made to deal with the fact that some preliminaries were held to late. Typically, it takes about five weeks to complete the logistic operations of an ICPC regional contest. Therefore, all preliminaries were held before the end of Octobor, and the regional contests were held in November in the past few years.

The 2023 ICPC Taoyuan Regional Contest will be held from Octobor 21 to Octobor 23, 2023. Consequently, some categories will be suspended if the corresponding preliminaries are still held in late September or Octobor as usual. As the director of TOPC, you must be wise enough to choose a proper date to hold TOPC. If the categories related to TOPC are suspended, there will be much less domestic teams competing in the 2023 ICPC Taoyuan Regional Contest. A contest date is too late for TOPC if it is not at least 35 days prior to Octobor 21, 2023. Please write a program to determine whether a tentative contest date is too late for TOPC.

Input Format

The input contains a string in ISO 8601 format `YYYY-MM-DD` indicating the tentative contest date for TOPC. `YYYY`, `MM`, `DD` are the year, the month, the day of the tentative contest date for TOPC, respectively.

Output Format

If the tentative contest date is too late, output "`TOO LATE`" without quotes. Otherwise, output "`GOOD`" without quote.

Technical Specification

- `YYYY` is 2023.
- `MM` is from 1 to 12.
- `DD` is from 1 to 28.

Sample Input 1

```
2023-09-16
```

Sample Output 1

```
GOOD
```

Sample Input 2

```
2023-10-01
```

Sample Output 2

```
TOO LATE
```

Problem B
Better Chance

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

In recent years, the ICPC community has expanded its global presence, with annual participation encompassing 111 countries, involving 3,450 universities, and engaging 75,000 team members, coaches, and volunteers. However, it is important to note that only approximately 130 teams earn the opportunity to advance to the annual ICPC World Finals. Specifically, for the 2023 ICPC World Finals, only 16 teams from the Asia Pacific Region have qualified. In the 2022 ICPC Asia Pacific Regional Contests, an impressive turnout saw over 1,800 teams representing 283 universities. Regrettably, the vast majority of these teams, more than 99%, were unable to progress beyond the regional contests.

The ICPC Asia Pacific Region has introduced the Asia Pacific playoff as a new tier in the selection process for teams to participate in the 2024 ICPC World Finals. This playoff represents a higher level of competition compared to the regional contests. In 2024, the first Asia Pacific playoff will take place in Hanoi, Vietnam. Approximately 66 teams will earn their spots in this playoff based on the rules outlined in the ICPC Asia Pacific Rules section. This expansion ensures that a greater number of teams will have opportunities to engage in higher-level contests in the coming years.

Due to the introduction of ICPC Asia Pacific Playoff contest, the ICPC Asia Pacific Rules have been changed greatly. After reading the new rules, you find that you are allowed to participate two regional contests in 2023 to have a better chance to advance to the playoff. Your team plans to participate the ICPC Taoyuan Regional Contest and the ICPC Jakarta Regional Contest in 2023. Assume that your team's *recomputed team ranks* and the *site scores* of both regional contests are given. Please write a program to determine in which regional contest your team has a better chance of advancing.

(Note: The definition of *recomputed team rank* and *site score* can be found in the ICPC Asia Pacific Rules section.)

Input Format

The first line of the input contains four space-separated numbers R_T, R_J, S_T, S_J. R_T and R_J are your team's *recomputed team ranks* of Taoyuan Regional and Jakarta Regional, respectively. S_T and S_J are the site scores of Taoyuan Regional and Jakarta Regional, respectively.

Output Format

If your team has a better chance of advancing in Taoyuan Regional, output `TAOYUAN`. If your team has a better chance of advancing in Jakarta Regional, output `JAKARTA`. If the chances are equally good, output `SAME`.

Technical Specification

- R_T and R_J are positive integers less than 66.
- You may assume that your team ranks are in the top 50% in both regional contest.
- $1 \leq S_T \leq 2000$ and $1 \leq S_J \leq 2000$.
- $100 \cdot S_T$ and $100 \cdot S_J$ are positive integers.

Sample Input 1

```
1 2 34.56 56.78
```

Sample Output 1

```
TAOYUAN
```

Sample Input 2

```
2 3 45.67 98.01
```

Sample Output 2

```
JAKARTA
```

Sample Input 3

```
4 5 33.33 44.44
```

Sample Output 3

```
SAME
```

ICPC Asia Pacific Rules

The following rules do not apply to teams from South Pacific (Australia, New Zealand, etc.). World Finals teams will be selected from the South Pacific Independent Regional Contest independently of the other Asia Pacific regionals.

Basic rules of Asia Pacific regionals

A1. A team can participate in at most two regionals.

A2. If a regional is accompanied with preliminary contests, teams from the hosting country of the regional must be qualified for the regional through the corresponding preliminary contests.

A3. Regionals are expected to accept teams from other countries in Asia Pacific. Each regional may set an upper-bound of the number of foreign teams. The director of the regional may define appropriate rules for selecting the foreign teams to be accepted.

A4. Regionals may accept teams from South Pacific as well as other super-regions such as Asia East. These teams are never qualified for the World Finals through Asia Pacific regionals.

A5. Teams from a country hosting a regional should not compete in two regionals in other countries.

Selection rules for the World Finals (part 1)

B1. The winner team of each regional is directly qualified for the World Finals. Here, "winner" is defined as the team with the highest rank after excluding teams from other super-regions or South Pacific.

B2. If two or more teams from a single university are winners of different regionals, only one of them is qualified for the World Finals. The university should select one of them. Optionally, the university can let the teams compete in the playoff to select one of them for the World Finals. (These teams are invited to the playoff. See D1.)

B3. If one of the winner teams from a single university is to be selected in the playoff (see B2), the team with the highest rank among them in the playoff is qualified for the World Finals, regardless of the ranking of teams from other universities in the playoff.

Site scores of regionals

C1. For each regional, its "site score", say S, is defined as follows.

$$\begin{aligned} S \;=\; & 0.56 \times \text{number of universities in the regional} \\ & + 0.24 \times \text{number of teams in the regional} \\ & + 0.14 \times \text{number of universities in the preliminary contests} \\ & + 0.06 \times \text{number of teams in the preliminary contests} \\ & + 0.30 \times \text{number of foreign teams in the regional} \end{aligned}$$

Here, only the teams solving at least one problem are counted. Teams from South Pacific and other super-regions are counted (not excluded).

(Note) The above formula is copied from the World Finals team selection rules in Asia Pacific since 2020.

Selection rules for the playoff

D1. The winner teams of regionals are invited to the playoff as open participation teams. Their performance in the playoff does not affect their qualification for the World Finals. There is an exception if the teams are competing in the playoff due to the rules B2 and B3.

(Note) From the winner universities, only the winner teams can compete in the playoff. Other teams (second or lower in the regional ranking) are not allowed to compete.

(Note) Winner teams are included in the playoff ranking, but are removed in E1(1) below, not affecting the qualification of teams from other universities.

D2. Apply the following procedures in this order, to the ranking of each regional. (The regional is called X in the following.)

(1) Remove teams from South Pacific and other super-regions from the ranking.

(2) Remove teams not solving any problem from the ranking. Only teams solving at least one problem are retained.

(3) Keep top 50% teams (rounded up) after the previous step, and remove teams ranked lower than this. This means that lower 50% teams are never qualified for the playoff.

(Note) The intent of step (3) is to never allow teams in the lower half ranking of

a regional to compete in the playoff. This step is effective only in rare cases. For example, teams from under- represented countries will not be qualified unless it gets a rank in the top half. See D3(3).

(4) Remove the teams of winner universities from the ranking. Here, a winner university is the university to which a winner team (see B1) belongs. It may be the winner of the regional X, or of another regional.

(5) Remove the fourth or lower ranked teams of each university from the ranking.

(Note) The maximum number of teams from a single university competing in the playoff is 3. See D3(2).

(6) Recompute the team rank of each remaining team afresh. Let R be the recomputed team rank of a team. Assign the following value to each team in the ranking.

$$(R–1)/S$$

S is the site score (defined in C1 above) of the regional X.

(Note) This definition and the following steps are based on Shieh-Ishihata formula. It was the main part of the World Finals team selection rules in Asia Pacific, before COVID-19.

D3. Merge the lists of teams from all regionals in Asia Pacific, and apply the following procedures in this order to the resulting list.

(1) Sort the resulting list in ascending order of the value assigned above.

(2) Strike out the second instance of a single team, if any. Then, strike out the fourth or later instances of teams of a single university. This means that at most three teams are qualified from a single university.

(Note) The fourth or lower teams are already removed in D2(5), but this step is necessary.

(3) From each Asia Pacific country, select one team with the smallest value. This team is qualified for the playoff.

(Note) This is a wild-card rule for under-represented countries. At least one team is qualified from every country. However, there is a condition. The team must be in the top half rank of a regional. See D2(3).

(4) Let P be the number of teams for the playoff. Scan the list from the team with the smallest value, and select teams one by one, skipping those teams already selected in the step (3), unitl the number of selected teams reach P. These teams are qualified for the playoff.

D4. When an additional team should be selected, that is, when a qualified team declines to

participate in the playoff, reapply the rule D2 (and consequently D3) after removing that team from the ranking of all regionals.

(Note) In the case of D4, site scores are not recomputed.

(Note) There will be no penalty to the teams declining to compete in the playoff.

Selection rules for the World Finals (part 2)

E1. World Finals teams are first selected according to B1 —B3 above. Then, the teams with the highest playoff ranks are selected, according to the following procedures.

(1) Strike out the teams with open participation status (winner teams of regionals) in the ranking of the playoff.

(2) Strike out the second or lower ranked teams of a university in the ranking of the playoff.

(3) Let N be the number of the World Finals slots allocated to Asia Pacific. Let M be the number of teams qualified for the World Finals in B1 —B3. Then, top N –M teams of the modified ranking are qualified for the World Finals.

Almost blank page

Problem C
Cutting into Monotone Increasing Sequence

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

A monotone sequence is a sequence of numbers that either consistently increases or consistently decreases as you move along the sequence. In other words, it exhibits a consistent trend in either an upward or downward direction.

In a monotone increasing sequence, each term in the sequence is greater than or equal to the preceding term. Mathematically, for a sequence $a_1, \ldots, a_n$, it is monotone increasing if and only if for every $1 \leq i < n$, $a_i \leq a_{n+1}$. For example, the sequence $1, 2, 2, 4, 5$ is a monotone increasing sequence because each term is greater than or equal to the previous term.

Monotone sequences are important in various areas of mathematics, including calculus and analysis, as they often simplify the analysis of functions and their behavior. They provide a clear and consistent trend that makes it easier to understand the behavior of a sequence or a function over a range of values.

One of our problem setters is fond of big integers. Over the past few years, the Taiwan Online Programming Contest has frequently featured problems involving big integers. This time, we have a problem that combines big integers with monotone increasing sequences. Your task is to transform a big integer, denoted as x, into a monotone increasing sequence by inserting commas '`,`' into the gaps between its digits, while adhering to following constraints.

- The last term of the monotone increasing sequence is no more than b.
- Commas cannot be inserted before a zero.
- The number of commas is minimized.

Let's assume that x is an integer with k digits and is represented as $d_1 d_2 \cdots d_k$. For instance, if we have $x = 654321 = d_1 d_2 \cdots d_6$ and $b = 1000$, we can insert commas into gaps after d_3 and d_5 to convert x into the following monotone increasing sequence: $6, 54, 321$.

Please write a program to compute the minimum number of commas required to transform a given big integer x into a monotone increasing sequence consisting of numbers no more than a given integer b. If there is no way to transform, please print '`NO WAY`'.

Input Format

The input contains two non-negative integers x and b.

Output Format

Print the minimum number of commas required to transform x into a monotone increasing sequence consisting of numbers no more than b. If there is no way to transform, please print

'NO WAY' without quotes.

Technical Specification

- $x \leq 10^{100000}$
- $b < 2^{64}$
- There is no leading zero if $x > 0$.

Sample Input 1

```
654321 1000
```

Sample Output 1

```
2
```

Sample Input 2

```
654321 100
```

Sample Output 2

```
NO WAY
```

Problem D
Divide a Convex

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

A polygon is a geometric shape that consists of a closed set of straight line segments connected end-to-end to form a closed loop. These line segments enclose a region of space called the polygon's interior. The points where the line segments meet are called *vertices*, and the line segments themselves are called *edges*.

A simple polygon is a polygon that has no self-intersecting edges and has no holes. In other words, none of its edges cross over or intersect with each other within the interior of the polygon, and at each of its vertex, exactly two of its edges meet.

A convex polygon is a specific type of simple polygon that has an additional properties: All interior angles are strictly less than 180 degrees. In a convex polygon, if you were to draw straight lines connecting any two points inside the polygon, those lines would always remain inside the polygon.

David has a land with a convex polygon shape that has n vertices $(x_1, y_1), \ldots, (x_n, y_n)$. He wants to divide the land into two parts by a line segment $\overline{PQ}$ satisfying the following criteria.

- P and Q are points located on different edges of the convex polygon to be divided.
- The two parts are convex polygons with an equal perimeter. That is, the sum of the lengths of the first part's edges equals the sum of the lengths of the second part's edges.

Please help David to find out the minimum length of $\overline{PQ}$.

Input Format

The first line contains an integer n indicating the number of vertices of the convex polygon to be divided. The i-th of the following n lines contains two space-separated integers x_i and y_i indicating that a vertex of the convex polygon is at the point (x_i, y_i).

Output Format

The length of the shortest line segment that divides the input convex polygon into two equiperimeter convex polygons.

Technical Specification

- $n \leq 100000$
- $x_i, y_i \in [-10^8, 10^8]$ for $1 \leq i \leq n$.
- There is no guarantee $\overline{(x_i, y_i)(x_{i+1}, y_{i+1})}$ will be an edge.
- The answer is considered correct if the precision error is less than 10^{-6}.

Sample Input 1

```
4
0 0
1 10
0 10
1 0
```

Sample Output 1

```
1
```

Sample Input 2

```
5
0 0
10 10
0 10
10 0
5 11
```

Sample Output 2

```
10.0
```

Problem E
Exponentiation

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Exponentiation is a mathematical operation that involves raising a base number to a certain exponent to obtain a result. In the expression a^n, where a is the base and n is the exponent, it means multiplying a by itself n times. The result of this operation is called the *exponentiation* of a to the n-th power. For examples, $2^3 = 2 \times 2 \times 2 = 8$ and $5^2 = 5 \times 5 = 25$. In these examples, 2 is the base, 3 is the exponent in the first case, and 5 is the base, and 2 is the exponent in the second case. Exponentiation is a fundamental operation in mathematics and is commonly used in various contexts, such as solving equations, and cryptography.

In many cryptographic algorithms, particularly those based on number theory like RSA (Rivest-Shamir-Adleman) and Diffie-Hellman, modular exponentiation is a fundamental operation. Modular exponentiation involves raising a base to an exponent modulo a modulus. This operation is computationally intensive but relatively easy to perform, even for very large numbers.

Let $x + \frac{1}{x} = \alpha$ where α is a positive integer. Please write a program to compute $x^\beta + \frac{1}{x^\beta} \bmod m$ for given positive integers β and m.

Input Format

The input has only one line, and it contains three space-separated positive integers α, β and m.

Output Format

Outout $x^\beta + \frac{1}{x^\beta} \bmod m$. If there are multiple solutions, you may output any of them in the range from 0 to $m - 1$.

Technical Specification

α, β and m are positive integers less than 2^{64}. You may assume $x^\beta + \frac{1}{x^\beta}$ is an integer.

Sample Input 1

```
1 2 3
```

Sample Output 1

```
2
```

Sample Input 2

```
5 4 321
```

Sample Output 2

```
206
```

Sample Input 3

```
3 3 333
```

Sample Output 3

```
18
```

Sample Input 4

```
8 8 888
```

Sample Output 4

```
626
```

Note

x can be a complex number. For example, x is either $\frac{1+\sqrt{3}i}{2}$ or $\frac{1-\sqrt{3}i}{2}$ if $\alpha = 1$. However, $x^\beta + \frac{1}{x^\beta}$ is always an integer in this problem.

Problem F
Finding Bridges

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

An undirected graph is a fundamental concept in graph theory, a branch of mathematics and computer science. It is a data structure that represents a set of vertices connected by edges that have no direction. In other words, in an undirected graph, if there is an edge from vertex u to vertex v, there is also an edge from vertex v to vertex u. We use a two-element set $\{A, B\}$ to represent such undirected edge. An undirected graph is simple if there is at most one edge between any pair of vertices.

In graph theory, a connected component is a group of vertices and edges within a graph where you can travel from any vertex to any other vertex by following a sequence of edges. A bridge is an edge in an undirected graph that, if removed, increases the number of connected components in the graph.

Bridges are essential concepts in graph analysis and have practical applications in network design, fault tolerance, and connectivity analysis. They are often used to identify vulnerable points in a network where the removal of a single edge could lead to the isolation of certain components or the disruption of connectivity. Identifying bridges in a graph can be done using various algorithms which can detect these crucial edges and help analyze the robustness and structure of a network or system.

You have a simple undirected graph whose vertices and edges are $V = \{1, 2, \ldots, n\}$ and $E = \{\{u_1, v_1\}, \ldots, \{u_m, v_m\}\}$. Your friend, Flora, ask you to sequentially remove edges $e_1, e_2, \ldots, e_q$ from your graph and report the number of bridges left in the graph after each removal. Please write a program to generate the report.

Input Format

The first line contains three space-separated non-negative integers n, m and q. n is the number of vertices of the graph. m is the number of edges of the graph. q is the number of edges to be removed. The i-th of the following m lines contains two space-separated positive integers, u_i and v_i, indicating the i-th element of E. Then q lines follows. The j-th of them contains two space-separated positive integers, x_j and y_j, indicating the j-th removed edge e_j is $\{x_j, y_j\}$.

Output Format

Output q lines. The k-th line should contain an integer b_k indicating the number of bridges in the graph after removing k edges.

Technical Specification

- $1 < n \leq 200000$

- $1 \leq m \leq \min\left(200000, \binom{n}{2}\right)$
- $1 \leq q \leq m$
- There are two ways to represent an edge $\{u, v\}$: "u v" and "v u".
- $\{x_i, y_i\}$ must be an edge in E.
- No edge will be removed twice.

Sample Input 1

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

Sample Output 1

```
0
2
1
3
2
```

Problem G
Gadget Construction

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

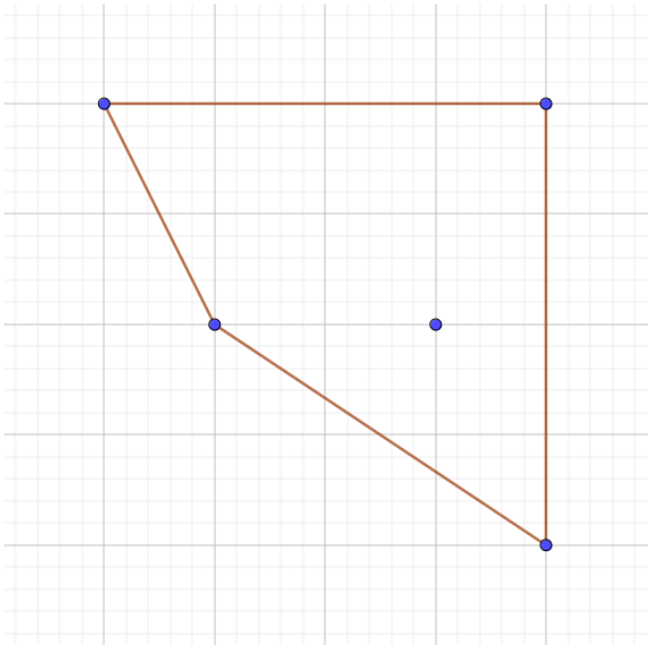
Grigory is a talented engineer who has a love in developing drones and, of course, geometry. As a skilled problem solver, he is able to come up with solutions in difficult situations, but today, he encountered a problem that his abilities alone are not enough to solve. Therefore, as his best sidekick, your task is to assist him.

There are n cogs on the Cartesian plane. The i-th cog is located at the coordinates (x_i, y_i) and has a character c_i that describes its color – "`b`" means it is a black cog, while "`w`" means it is a white cog. In addition, no three cogs lie on the same line.

Grigory wants to build a gadget using some cogs. To do this, he first chooses a subset S that consists of 4 or more of the n cogs. Then, a loop of chains is placed around the cogs. The loop is chosen such that:

- Every cog in S either touches the loop or lies in its interior.
- The total length of chains is minimal.

For example, the image below shows the chains that would be placed for a chosen set of cogs.



Finally, consider the cogs in S that touch the loop. These are the most important cogs, so they cannot be interfering with each other. Therefore, any two adjacent cogs on the loop must have different colors, otherwise the resulting gadget won't be working properly. If a cog in S does not touch the loop, then it may have an arbitrary color.

How many different sets S can Grigory choose to build a properly working gadget? Since the answer can be very large, find the value modulo $10^9 + 7$.

Input Format

The first line contains an integer n. Then n lines follow. The i-th line contains two integers x_i, y_i and a character c_i denoting the coordinates and color of the i-th cog.

Output Format

Print the number of sets S that satisfy the conditions modulo $10^9 + 7$.

Technical Specification

- $4 \leq n \leq 500$
- $-10^8 \leq x_i, y_i \leq 10^8$ for $i = 1, 2, \ldots, n$
- $c_i \in \{\texttt{b}, \texttt{w}\}$ for $i = 1, 2, \ldots, n$
- $(x_i, y_i) \neq (x_j, y_j)$ for $i \neq j$
- No three cogs lie on the same line.

Sample Input 1

```
4
0 0 w
1 0 b
1 1 w
0 1 b
```

Sample Output 1

```
1
```

Sample Input 2

```
6
0 0 w
0 4 b
1 2 w
3 2 b
4 0 b
4 4 w
```

Sample Output 2

```
9
```

Sample Input 3

```
4
0 0 b
1 0 w
1 1 w
0 1 b
```

Sample Output 3

```
0
```

Problem H
Heap Structure

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

A minimum binary heap, often simply referred to as a "min-heap," is a specialized type of binary tree-based data structure used in computer science. It is a binary tree that has two main properties:

1. Heap Property: In a min-heap, for any given node, the value of that node is less than or equal to the values of its child nodes. This means that the minimum value in the heap is always found at the root node. In other words, the smallest element in the heap is at the top.

2. Binary Tree Structure: A min-heap is typically a complete binary tree, which means that all levels of the tree are fully filled except possibly for the last level, which is filled from left to right. This structural property allows min-heaps to be efficiently implemented using arrays.

Min-heaps are often used to implement priority queues, which are data structures that maintain a collection of elements with associated priorities. By keeping the minimum element at the root, min-heaps can quickly retrieve and remove the element with the highest priority (lowest value) in logarithmic time. However, removing any other element from a min-heap may need linear time.

Hank recently learned min-heaps. He wonders how many nodes can keep the k-th smallest element in a min-heap of n-nodes. Please write a program to help him.

Input Format

The input contains two space-separated positive integers n and k.

Output Format

Output the number of nodes that can keep the k-th smallest element in a min-heap of n nodes.

Technical Specification

$1 \leq k \leq n \leq 10^{18}$

Sample Input 1

```
100 1
```

Sample Output 1

```
1
```

Sample Input 2

```
12 3
```

Sample Output 2

```
6
```

Problem I
Introversion

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

You run a restaurant called Taste Of Pacific Cuisine (TOPC). This weekend, you will be hosting a large banquet that caters a sizable group of guests. Your chef designed n types of dishes. To ensure every guest a chance to taste each type of the dishes, you plan to prepare **two dishes** per dish type. (Hence there are a total of $2n$ dishes at the banquet.)

There is a very long table in your restaurant, and you plan to exhibit all $2n$ dishes in a line on this table all at once. Not surprisingly, the length of the table fits exactly $2n$ dishes. As a considerate host, you plan to make sure that no two dishes of the same type are laying on the table together — this allows a variety of choices for introversion individuals who prefer not to wander around.

Now, some dishes have already been brought to the table. Can you quickly count the number of ways to place the remaining dishes such that no two dishes of the same type are placing together? You must compute this number quickly so you can give an introductory overview to your kitchen staff on how to place the remaining dishes — that's what you call an intro version. Since the number of ways could be large, it suffices to output the answer modulo $10^9 + 7$.

Input Format

The first line contains an integer T, denoting the number of test cases. For each test case, the first line contains an integer n. The second line contains $2n$ integers $x_1, x_2, \cdots, x_{2n}$ separated by a space. If $x_i = 0$ it means that the i-th position on the table is empty. Otherwise, x_i will be an integer ranges between 1 and n denoting the type of the dish. It is guaranteed that for each dish type $k \in \{1, 2, \ldots, n\}$, k occurs at most twice in the input sequence. In addition, if k does occur two times in the sequence, these two numbers will not be neighboring.

Output Format

Output the number of valid ways to serve all the remaining dishes, modulo $10^9 + 7$.

Technical Specification

- $1 \leq T \leq 20$.
- $2 \leq n \leq 100$.
- $0 \leq x_i \leq n$.

Sample Input 1

```
3
2
1 2 0 0
2
1 0 2 0
5
0 0 0 0 0 1 2 3 4 5
```

Sample Output 1

```
1
0
96
```

Problem J
Java Warriors

Time limit: 1 second
Memory limit: 2048 megabytes

Problem Description

Jerry has earned acclaim as a renowned coach in the highly competitive realm of the International Collegiate Programming Contest (ICPC). His coaching prowess is exemplified by his unique approach — meticulously training his students to excel in ICPC competitions by harnessing the power of Java, a programming language seldom utilized in this arena. Consequently, Jerry's teams have become synonymous with excellence and are affectionately known as the "Java Warriors."

Jerry coaches n teams to compete in the 2023 ICPC Taoyuan Regional Contest. The registration fee is 4000 dollars per team, and Jerry does not have enough funds to pay for all Java Warriors. Jerry, facing a financial challenge in coaching multiple teams for the 2023 ICPC Taoyuan Regional Contest, prays to God with a heartfelt and sincere prayer.

"Dear God,

I come before you today with a humble heart and a deep desire to help my teams compete in the international collegiate programming contest. The registration fee is a significant hurdle, and I find myself lacking the funds needed to support them all.

I pray for your guidance and assistance, not only for myself but for the talented students who have put their trust in me as their coach. Please provide us with the resources necessary to cover the registration fees for these teams.

I understand that this contest is not just about winning but about fostering learning, teamwork, and growth in these young minds. Help us to overcome this financial obstacle so that we can continue to nurture their skills and aspirations.

I also pray for the wisdom to make the best decisions and the determination to explore all available options to secure the funds needed. If it is your will, open doors for us, connect us with individuals or organizations willing to support our cause, or inspire creative solutions to meet our financial needs.

Thank you, God, for hearing my prayer and for being a source of strength and guidance. I place my trust in you, knowing that with your help, we can overcome this challenge.

In your holy and gracious name, I pray,

Amen."

Upon hearing his heartfelt prayer, deeply moved by his sincerity and dedication, you are inspired to extend your support by covering the registration fees for the Java Warriors. Now, the

question arises: how much should you donate to enable Jerry's teams to compete in the 2023 ICPC Taoyuan Regional Contest?

Input Format

The input contains a positive integer n indicating there are n teams of Jerry's Java Warriors registering for the 2023 ICPC Taoyuan Regional Contest.

Output Format

Output the total amount of registration fees.

Technical Specification

n is no more than 20.

Sample Input 1	Sample Output 1
1	4000

Problem K
Kick

Time limit: 3 seconds

Memory limit: 2048 megabytes

Problem Description

You are given a string s consisting of lowercase English letters. A "kick" is defined as a substring of s that starts with the letter 'k' followed by the letter 'i' followed by the letter 'c' followed by the letter 'k'.

Your task is to count the number of distinct "kicks" in the string s. Note that the kicks can overlap.

Input Format

The input contains exactly a string s consisting of lowercase English letters.

Output Format

Print the number of "kicks" on a line.

Technical Specification

The length of the string s is no more than 5×10^6.

Sample Input 1	Sample Output 1
kickickstartkicks	3

Sample Input 2	Sample Output 2
kickkickkickkick	4

Almost blank page

Problem L
Location, Location, Location

Time limit: 3 seconds
Memory limit: 2048 megabytes

Problem Description

The concept of "location, location, location" is a common phrase used in real estate and business to emphasize the importance of the physical location of a property or business. In real estate, it suggests that the desirability and value of a property are heavily influenced by its location, often more so than by the property's features or condition. In business, it highlights that the success of a retail store or commercial establishment can be significantly impacted by its geographical location.

In recent years, people tend to buy electric vehicles, but the buyers soon find it is hard to install charging piles in their apartments or houses. Building a charging station can be a good idea to make a lot of money. Your boss, Lena, ask you to find a good location for establishing her charge station to serve the electric vehicle owners.

In recent times, there has been a growing trend towards the adoption of electric vehicles (EVs). However, many EV owners face the challenge of installing charging infrastructure in their apartments or houses. Establishing a charging station presents a lucrative opportunity in response to this demand. Your boss, Lena, has tasked you with identifying an optimal location for building a charging station to cater to the needs of electric vehicle owners.

You are given a list of n locations represented as (x, y) coordinates in a 2D plane. For each location, there is an apartment or a house without any charging infrastructure. Your task is to build a charging station at the location that is closest to all n locations on the list. In this problem, distance is measured using the Manhattan distance metric. The Manhattan distance between two points (x_1, y_1) and (x_2, y_2) is defined as $|x_1 - x_2| + |y_1 - y_2|$. Your goal is to find a location (x, y) such that the sum of the Manhattan distances from that location to all n locations on the list is minimized.

Input Format

The first line contains a positive integer n indicating the number of locations. The i-th of the n following lines contains two integers x_i and y_i. The coordinates of the i-th location is (x_i, y_i).

Output Format

Print the location (x, y) minimizing the sum of Manhattan distance from (x, y) to all n locations on the list. If there are multiple solutions, output the one minimizing x. If there still multiple solutions, output the one minimizing y.

Technical Specification

- $1 \leq n \leq 100000$

- $-100000 \leq x_i \leq 100000$ for $1 \leq i \leq n$.
- $-100000 \leq y_i \leq 100000$ for $1 \leq i \leq n$.

Sample Input 1

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

Sample Output 1

```
1 1
```

Sample Input 2

```
2
0 0
2 2
```

Sample Output 2

```
0 0
```