



Feng Shui House (fengshui)

Giorgio recently became very passionate about *Feng Shui*, the art of *harmonizing everyone with the surrounding environment*, and decided to rebuild his home from scratch in order to optimize his harmony with the universe. First, he searched his yard for geomagnetic nodes, producing a list of N points with coordinates (X_i, Y_i) . Now, he needs to select four of those points such that together they will form a perfect square whose sides are aligned with the cardinal axes. These points will determine the perimeter of his new home.

However, it is not easy to examine such a long list and as of now Giorgio has only found few very small valid squares, not really fit to contain a whole house. Help Giorgio find the largest square that obeys the *Feng Shui* principles!

🔗 Among the attachments of this task you may find a template file `fengshui.*` with a sample incomplete implementation.

Input

The first line contains the only integer N . The subsequent N lines contain two integers X_i, Y_i each.

Output

You need to write a single line with an integer: the maximum side length for a Feng Shui compliant house.

Constraints

- $4 \leq N \leq 50\,000$.
- $0 \leq X_i, Y_i \leq 1000$ for each $i = 0 \dots N - 1$.
- There exists at least one Feng Shui compliant square.
- There are no repeated points in the list.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

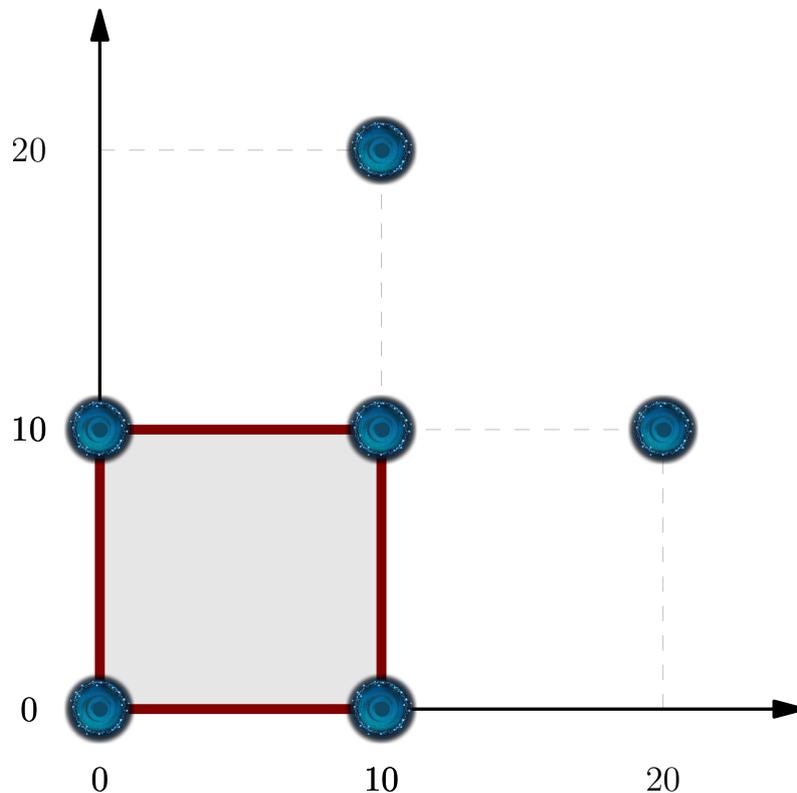
- **Subtask 1** [5 points]: Examples.
- **Subtask 2** [35 points]: $N \leq 50$.
- **Subtask 3** [30 points]: $N \leq 500$.
- **Subtask 4** [30 points]: No additional limitations.

Examples

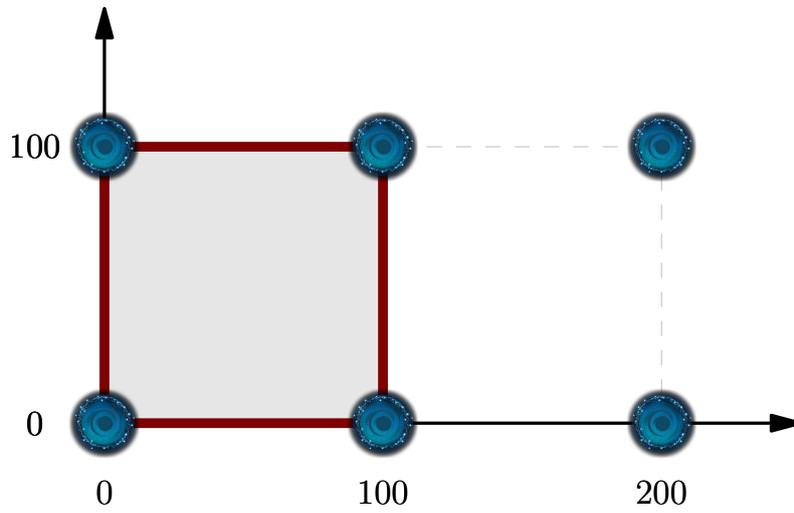
input.txt	output.txt
<pre>6 0 10 20 10 10 10 10 0 10 20 0 0</pre>	<pre>10</pre>
<pre>6 0 0 0 100 100 0 100 100 200 0 200 100</pre>	<pre>100</pre>

Explanation

In the **first sample case**, there is only one square aligned with the axes. Another larger square would be possible, but it would not be aligned with the axes.



In the **second sample case** there are two possible squares of the same size.



Gambling Assistant (gamble)

Giorgio has been recently introduced to a gambling club, where he is accurately losing all of his money. Tired of losing so much, he chooses one of the simplest games played there and tries his best to become good at it. This game consists in a series of N rounds, in which every player draws a card from a deck (represented by a number from 1 to 13). The hand size is limited to K cards, so that whenever a player exceeds this limit he has to immediately discard one card from his hand. During the game several bets take place, that are won by the player with the highest grand total (sum of the values – from 1 to 13 – of each card) in his hand.



Figure 1: The typical hand picked by Giorgio, featuring a grand total of 4.

To improve his performance in the game, Giorgio is reviewing some videos of famous plays of this game. In this videos he is able to see which card C_i is drawn at each round, but he is not able to easily keep track of the grand total as it varies during the game. Help Giorgio in determining the *highest* grand total G_i that a player can have at each round, given the list of cards C_i drawn!

 Among the attachments of this task you may find a template file `gamble.*` with a sample incomplete implementation.

Input

The first line contains the two integers N and K . The second line contains N integers C_i .

Output

You need to write a single line with N integers G_i , the highest grand total possible when drawing the given cards.



Constraints

- $1 \leq K \leq N \leq 100\,000\,000$.
- $1 \leq C_i \leq 13$ for each $i = 0 \dots N - 1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** [5 points]: Examples.
- **Subtask 2** [10 points]: $N = K$.
- **Subtask 3** [10 points]: $K = 1$.
- **Subtask 4** [30 points]: $N \leq 100$.
- **Subtask 5** [25 points]: $N \leq 10\,000$.
- **Subtask 6** [20 points]: No additional limitations.

Examples

input.txt	output.txt
4 2 10 4 10 12	10 14 20 22
8 5 4 13 10 2 7 1 13 7	4 17 27 29 36 36 47 50

Explanation

In the **first sample case**, the hands kept by the player (in order) are:

[10] [4, 10] [10, 10] [10, 12].

In the **second sample case**, the hands kept by the player (in order) are:

[4] [4, 13] [4, 10, 13] [2, 4, 10, 13] [2, 4, 7, 10, 13]
[2, 4, 7, 10, 13] [4, 7, 10, 13, 13] [7, 7, 10, 13, 13].

Interstellar Transmissions (seti)

After reading about the ambitious *SETI* program, William decided to join the project and build an array of N radios for interstellar transmissions in his garage. However, the array has not received any extraterrestrial message yet, fact that William blamed to the phenomenon of *interference*.

In particular, William noticed that whenever the i -th radio is turned on, V_i of the radios on its left (that is, radios $j = i - V_i \dots i - 1$) receive disturbed signals and are not usable. Fortunately, no interfering signals are received by radios on the right¹ (that is, with $j > i$). In order to plan his next experiment avoiding interferences, William now needs to select a subset of his radios avoiding interferences, that is, such that if radio i is turned on then all radios between $i - V_i$ and $i - 1$ are turned off.



Figure 1: An array of radio telescopes, pretty much alike the one in William's garage.

Help William plan his next experiment, by counting the number of subsets of radios avoiding interferences modulo² 1 000 000 007.

 Among the attachments of this task you may find a template file `seti.*` with a sample incomplete implementation.

Input

The first line contains the only integer N . The second line contains N integers V_i .

Output

You need to write a single line with an integer: the number of valid subsets of radios modulo 1 000 000 007.

¹This phenomenon is due to the peculiar disposition and frequency arrangement chosen by William. There is a truly marvelous description of this arrangement, which this margin is too narrow to contain.

²The modulo operator is `%` in C/C++ and `mod` in Pascal.

Constraints

- $1 \leq N \leq 1\,000\,000$.
- $0 \leq V_i \leq i$ for each $i = 0 \dots N - 1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1 [5 points]:** Examples.
- **Subtask 2 [20 points]:** $V_i = 1$ for each $i = 0 \dots N - 1$.
- **Subtask 3 [30 points]:** $N \leq 10$.
- **Subtask 4 [25 points]:** $N \leq 1000$.
- **Subtask 5 [20 points]:** No additional limitations.

Examples

input.txt	output.txt
3 0 0 0	8
6 0 1 2 3 2 1	13

Explanation

In the **first sample case**, there is no interference thus all 8 subsets of the three radios are valid.

In the **second sample case**, the 13 valid subsets are the following:



Public Transport (teleport)

After solving the black hole equations, Giorgio has patented a brand-new project for a *teleport machine*. This handy device would approximately resemble a photo booth, and allow customers to be instantly dematerialized and then fully reassembled into a perfectly functional human being, in another teleport machine of their choice. Since there are no known side effects as of now, Giorgio is enthusiastic about his new project and cannot wait to realize it. However, building a teleport machine is pretty expensive (and definitely out of his budget): so it is time to get a sponsor!

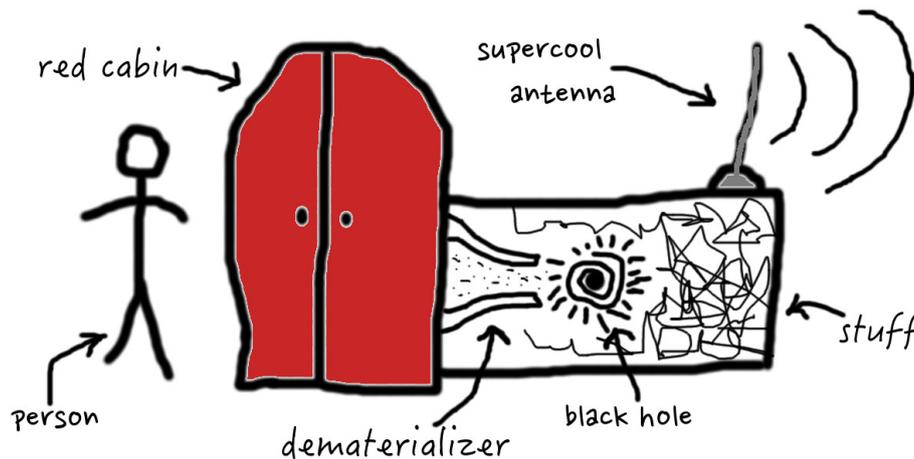


Figure 1: An excerpt of Giorgio's patented project.

Giorgio laid out a map of the city of Torino, consisting of streets, buildings and the proposed teleport machines. This map is a grid of $H \times W$ characters C_{ij} , which can be either:

- ‘.’ to represent a street, square or any other open space in the city;
- ‘#’ to represent a building, river or any other obstacles;
- ‘@’ to represent a teleport machine;
- ‘C’ to represent the city hall;
- ‘M’ to represent the mayor's house.

A citizen can freely move from a block to any adjacent block in any of the four directions N – E – W – S, unless the destination block contains an obstacle ‘#’. This operation takes 1 minute. If he is in a block containing a teleport machine, he can also use it to move to any other teleport machine in 1 minute.³

In order to persuade Chiara the city mayor to fund his project, it is crucial to calculate the time needed to go from the city hall to the mayor's house. In this way, he can show her how much it would be shortened by the teleport machines.

Among the attachments of this task you may find a template file `teleport.*` with a sample incomplete implementation.

³This is the time needed for paying a small fee and selecting the correct destination: the teleport itself is instantaneous.



Input

The first line contains the two integers H and W . Other H lines follow, each containing a string consisting of W characters among '.', '#', '@', 'C', 'M'.

Output

You need to write a single line with an integer: the duration in minutes of the fastest trip from the city hall to the mayor's house, possibly using the teleport machines.

Constraints

- $1 \leq H, W \leq 1000$.
- $C_{i,j}$ is a character among '.', '#', '@', 'C', 'M'.
- There is exactly one city hall and one mayor's house.
- There is always a path reaching the mayor's house starting from the city hall.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1 [5 points]**: Examples.
- **Subtask 2 [15 points]**: There are no obstacles ('#').
- **Subtask 3 [20 points]**: There are no teleport machines ('@').
- **Subtask 4 [20 points]**: $H, W \leq 10$.
- **Subtask 5 [20 points]**: $H, W \leq 100$.
- **Subtask 6 [20 points]**: No additional limitations.

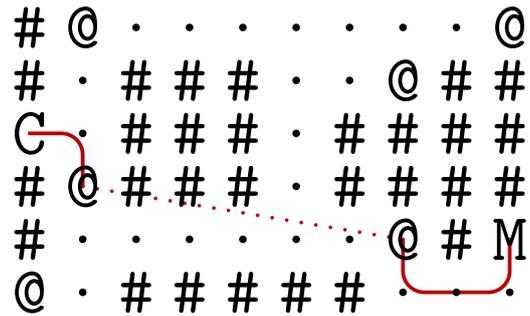
Examples

input.txt	output.txt
5 4 @..# ##.. C..M .#.# .#@.	3
6 10 #@.....@ #.###..@## C.###.#### #@###.#### #.....@#M @.#####...	7

Explanation

In the **first sample case**, it is not convenient to use the teleport machines.

In the **second sample case**, the best path is the following:





Confusing Usernames (usernames)

William spends most of his time administering the Italian portal for the IOI training. During his last sleepless night debugging the system, he figured out that several of the usernames used in the website are very similar. This causes a lot of confusion, especially when inspecting users' rankings.

He is then considering to add a new rule to the website: *you are not allowed to select a username whose set of letters is a subset of the set of letters of another username*. For example, you cannot register the username `bob00` if `n0ob` is already present. In order to evaluate the impact of such a design choice, he now wants to measure how much this rule is violated by the current list of usernames.

More precisely, given a list of N usernames U_i consisting of characters 'a' – 'z' and '0' – '9', he wants to count the number of *username pairs* (i, j) such that $\text{letters}(U_i) \subseteq \text{letters}(U_j)$.

 Among the attachments of this task you may find a template file `usernames.*` with a sample incomplete implementation.

Input

The first line contains the only integer N . The subsequent N lines contain the usernames U_i .

Output

You need to write a single line with an integer: the number of pairs violating the rule.

Constraints

- $1 \leq N \leq 10\,000$.
- Each username U_i is a string of length at most 60 consisting of characters 'a' – 'z' and '0' – '9'.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** [5 points]: Examples.
- **Subtask 2** [35 points]: $N \leq 100$ and there are no two usernames with the same set of letters.
- **Subtask 3** [25 points]: $N \leq 1000$.
- **Subtask 4** [25 points]: Usernames U_i consists only of characters 'a' – 'b'.
- **Subtask 5** [10 points]: No additional limitations.



Examples

input.txt	output.txt
3 carole rollercar 4ndr31	2
4 robin tyrionboss bornin2000 toy	3

Explanation

In the **first sample case**, `carole` and `rollercar` share the same set of letters, while `4ndr31` is unrelated. Thus the invalid pairs are (1, 2) and (2, 1).

In the **second sample case** the invalid pairs are (1, 2), (1, 3), (4, 2).

Scenic Walkway (walkway)

The managers of Gardaland Theme Park are building a new attraction, consisting of a sequence of N chambers of horrors, each located at a different height of H_i meters. Together with the attraction, they also plan to build a scenic walkway following most of the chambers from the outside.

In order to maximise the visibility of the new attraction, they need to carefully plan the altitude at which to build the walkway. For this reason they hired Giorgio and William, who calculated that it would be best if at least K chambers were clearly visible from the walkway. Given a set of chambers, define its *spread* as the difference between the highest and the lowest chamber in the set. Find the smallest possible spread for a set of K chambers!

Among the attachments of this task you may find a template file `walkway.*` with a sample incomplete implementation.

Input

The first line contains the two integers N and K . The second line contains N integers H_i .

Output

You need to write a single line with an integer: the smallest possible spread for a set of K chambers.

Constraints

- $2 \leq K \leq N \leq 1\,000\,000$.
- $0 \leq H_i \leq 1\,000\,000$ for each $i = 0 \dots N - 1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

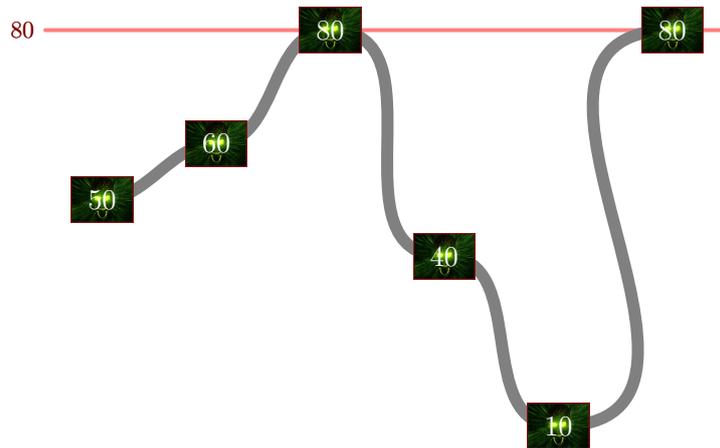
- **Subtask 1** [5 points]: Examples.
- **Subtask 2** [30 points]: $N \leq 10$.
- **Subtask 3** [25 points]: $N \leq 1000$, $K = 2$.
- **Subtask 4** [20 points]: $N \leq 1000$.
- **Subtask 5** [20 points]: No additional limitations.

Examples

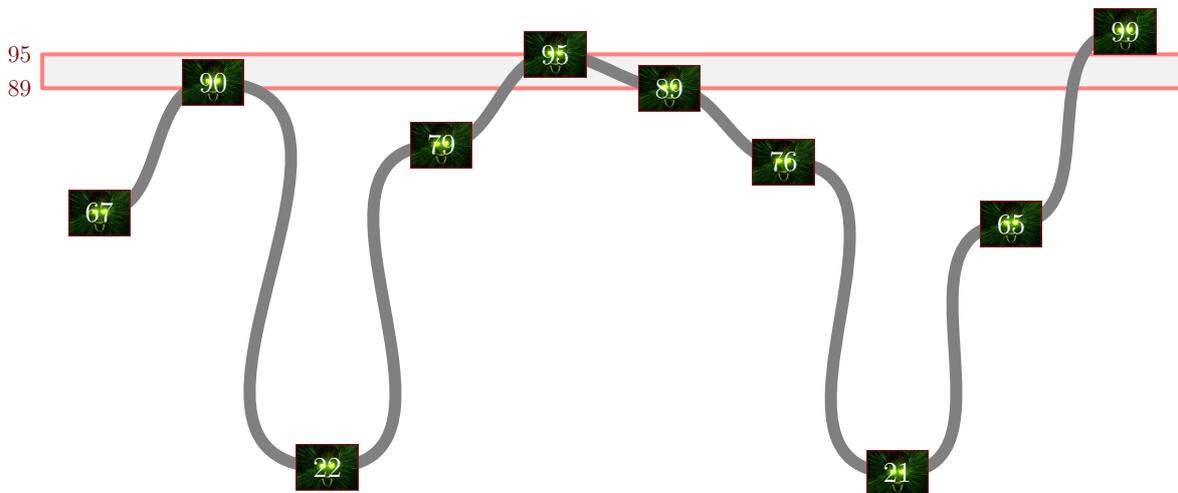
input.txt	output.txt
6 2 50 60 80 40 10 80	0
10 3 67 90 22 79 95 89 76 21 65 99	6

Explanation

In the **first sample case** the set of chambers with lowest spread is $\{80, 80\}$.



In the **second sample case** the set of chambers with lowest spread is $\{90, 95, 89\}$.



Water Calculator (water)

William has recently built a *Water Calculator*TM, which mimics the operation of a computer through water pipes, sinks and siphons. Due to the rudimentary nature of its circuits, the *Water Calculator*TM is able to store a single number up to a billion and perform only a handful of atomic operations:

- add or subtract 1;
- multiply by 2;
- divide by a power of 3 (which divides *exactly* the stored number).

These few operations are anyway more than sufficient for William's basic mathematical needs.

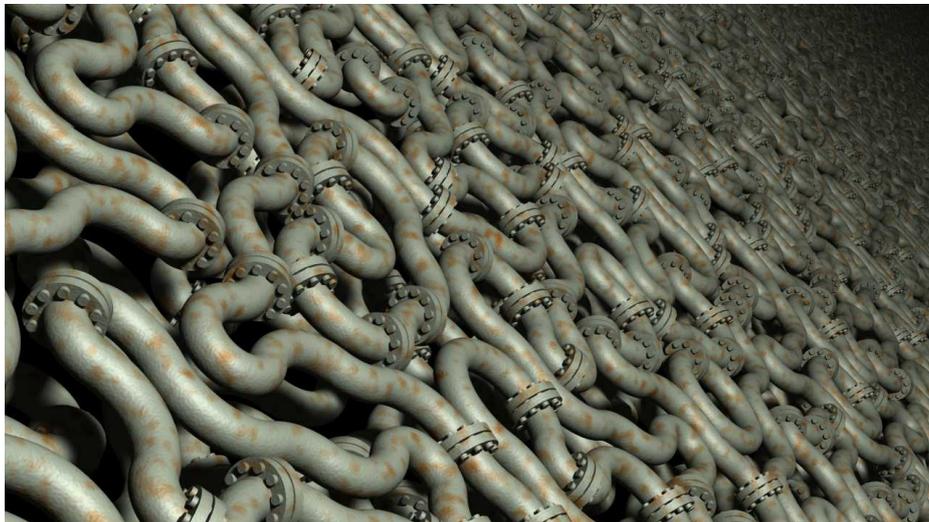


Figure 1: A small section of the *Water Calculator*TM.

In this moment, the *Water Calculator*TM is holding number N . Since William needs to leave his home for a while, it is crucial to empty the calculator in order to avoid malicious rusting! Calculate the minimum number of atomic operations necessary for William to completely empty the calculator.

👉 Among the attachments of this task you may find a template file `water.*` with a sample incomplete implementation.

Input

The first and only line contains the only integer N .

Output

You need to write a single line with an integer: the minimum number of atomic operations needed to empty the *Water Calculator*TM.



Constraints

- $1 \leq N \leq 10\,000\,000$.
- Operations that would increase the number stored over 10^9 are not allowed.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** [5 points]: Examples.
- **Subtask 2** [35 points]: $N \leq 10$.
- **Subtask 3** [30 points]: $N \leq 1000$.
- **Subtask 4** [30 points]: No additional limitations.

Examples

input.txt	output.txt
13	4
81	2

Explanation

In the **first sample case**, a possible sequence of operations is the following:

$$13 \rightarrow 26 \rightarrow 27 \rightarrow 1 \rightarrow 0$$

In the **second sample case**, a single operation brings 81 to 1 and a second one to 0.