Spring 2021 mBIT Standard Division

June 12, 2021

These problems are roughly ordered by difficulty. However, you should read and think about as many problems as you can in the time given. Good luck and happy coding!

Contents

Program Specifications

The memory limit for every problem is 256 MB.

Time Limits (seconds)			
Problem	**C++**	**Java**	**Python**
Mountain Climbing	1	1	1
Digit Sum	1	1	1
Reverse Race	1	1	1
Apple Orchard	1	1	1
Pokémon Permutation	1	1	1
Island Isolation	1	2	3
Map Matching	1	2	2
Street Layout	1	2	2
Grid Shuffling	1	1	2
Goomba Grouping	1	1	1
Rabbit Subtraction	2	2	3
Squid Art	1	2	2

Advice

Look at the pretests. You can access the first four pretests for each problem once you've made a submission. Some of the pretests are reduced in size to help you debug your program. Keep in mind that your final submission will be judged on a separate set of 40 hidden system tests for the official rankings.

Understand the new scoring system. Your program will only be submitted to the 40 system tests once it passes all 10 pretests. You will get one point for each system test you pass, **plus 20 points** if you get all of them correct (for a maximum of 60 points per problem). Results of the system tests will not be released until the end of the contest. Unlike last year, you will not get any points for programs that do not pass all pretests. Ties will be broken by the time of the last submission of a program which passes pretests.

Watch out for integer overflow. When a problem uses large values, make sure you use `long` (in Java) or `long long` (in C++). Python integers cannot overflow.

Use fast I/O. For problems with large input sizes, you may want to use faster I/O methods to prevent a time limit error. Here is how to use fast I/O in each language:

- In Python, write `from sys import stdin, stdout` at the top of your program. When reading input, use `stdin.readline()`. To write output, use `stdout.write()`.
- In Java, use a custom Scanner class as shown here.
- In C++, write `ios_base::sync_with_stdio(false); cin.tie(NULL);` at the top of your `main` method. Then you can use `cin` and `cout` as usual. Printing a single newline character (`\n`) is faster than `endl`.

Print extra digits for non-integer values in C++. If you are printing a `double` value in C++, by default it will only output a few digits (which may result in a wrong answer from our grader). To output real values with more precision, write `cout << setprecision(16);` at the top of your program. Our grader will accept real values in fixed **or** scientific notation (so `1234.56789`, `1.23456789E3`, and `1.23456789e+003` are treated the same). There will always be a tolerance for small relative errors between your solution and the correct answer.

Special considerations for Python. Make sure you're using fast I/O methods as described above. You can increase the recursion limit if your functions use repeated recursion. Additionally, we strongly recommend submitting your solutions in **PyPy**, which is typically faster than Python. Make sure you strip your input of trailing white space (this is especially important for our grader). Finally, `exit()` and `quit()` do not work with our grader.

Language versions. Our grader uses C++17 (compiled with the `-O3` tag), Java 11, and Python 3.9. Our version of PyPy implements Python 3.7.

Ask for clarifications! If you are confused about a problem statement, do not hesitate to message us.

THERE ARE THREE IMPOSTORS AMONG US.

See if you can find the hidden *Among Us* references in this contest! All three references are contained in problems shared between the Standard and Advanced divisions. Message us if you have found them all to receive a special shout-out!

§1 Mountain Climbing

Terraria's *creator, Andrew Spinks, is a big fan of* the Legend of Zelda *series. He incorporated many Zelda items and costumes into* Terraria.

Pablo is trying to return to his house in *Terraria*, but he must first cross a tall mountain. We can represent the mountain in two dimensions as a sequence of N adjacent columns with heights $a_1, \ldots, a_N$, measured in *Terraria* blocks. His path starts from the left base of the first column, crosses over the sides and tops of the columns, and ends at the right base of the last column (see the diagram). What is the total length of this path in blocks? Count both vertical and horizontal distance in your answer.

Input Format:

The first line contains N ($1 \leq N \leq 1000$).

The second line contains N integers $a_1, \ldots, a_N$ ($1 \leq a_i \leq 100$), the heights of the columns.

Output Format:

Output one line with the total length of the path.

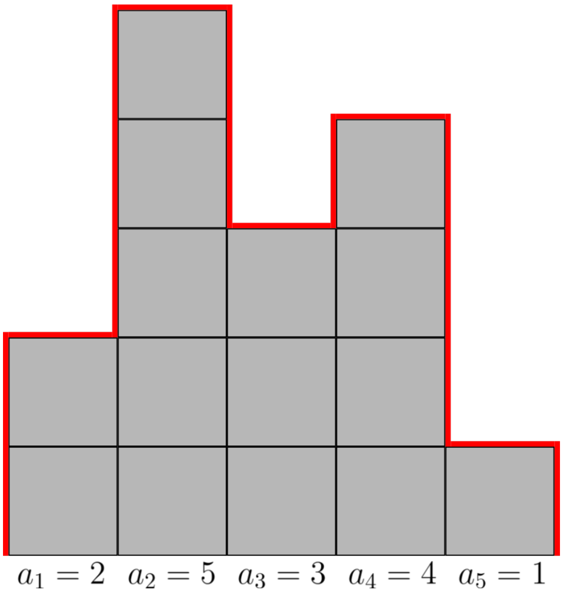
Sample Input:

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

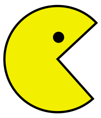
Sample Output:

```
17
```

The mountain is shown below, with the path in red. The path is 17 block lengths.



§2 Digit Sum



Pac-Man was originally named Puck Man, but he was changed to Pac-Man due to fears of people easily changing the P to an F.

Billy has just finished playing an intense game of Pac-Man. He claims that his score is M digits long, and that sum of these digits is N. Can you find any score that satisfies these conditions, or determine that no answer exists? Note that the number 0 counts as one digit.

Input Format:

The first and only line contains two integers N and M ($0 \leq N \leq 10^6$, $1 \leq M \leq 10^3$).

Output Format:

Output one line with an M-digit non-negative integer. It may not have any leading zeroes. If no such number exists, output `-1`.

If there are multiple correct answers, you may print any.

Sample Input:

```
11 3
```

Sample Output:

```
137
```

137 has 3 digits and its digit sum is $1 + 3 + 7 = 11$. Note that 137 is not the only answer, for example printing 425 would also work.

§3 Reverse Race

The highest typing speed ever recorded was 216 words per minute, set by Stella Pajunas in 1946.

The game *TypeRacer* has just released a new mode in which players are required to type backwards! They must type the words they see from right to left, reversing the order of the individual letters themselves. To complicate things, players must preserve capitalization: if a word in the original message is capitalized, the first letter of the reversed word must also be capitalized. For example, the message `I love Disney World` would become `Dlrow Yensid evol I`.

Kevin wants to code a bot to race in this mode. Can you help him determine the reversed message?

Input Format:

The first and only line contains a message of space-separated words. The first character of each word is an uppercase or lowercase English letter. The remaining characters of each word are all lowercase. The sum of the lengths of the words is between 1 and 1000 characters, inclusive.

Output Format:

Output the message with each word reversed.

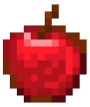
Sample Input:

```
I think the Montgomery Blair Informatics Tournament is so fun
```

Sample Output:

```
nuf os si Tnemanruot Scitamrofni Rialb Yremogtnom eht kniht I
```

§4 Apple Orchard



Bone meal can be used to accelerate the growth of plants in Minecraft.

Carlos wants to build an apple orchard in his *Minecraft* world. To start his orchard, he has determined that he needs at least A apples and at least B bones. He currently has N apples and M bones. It takes him X minutes of foraging to collect an apple, and Y minutes of fighting mobs to collect a bone.

In addition to foraging and fighting, Carlos can trade with the local villagers. One villager is offering an exchange of C apples for D bones, and another is offering D bones for C apples. Carlos may trade with either villager at any time and as often as he would like, as long as he is able to pay his side of the bargain. Completing a trade takes no time.

If Carlos plans out his actions optimally, what is the minimum number of minutes it will take him to get the necessary resources?

Input Format:

The first and only line contains eight integers N, M, X, Y, A, B, C, and D. All values are between 1 and 1000, inclusive.

Output Format:

Output one line with the answer.

Sample Input:

```
3 4 2 1 10 10 20 1
```

Sample Output:

```
7
```

Carlos starts with 3 apples and 4 bones, and he needs 10 apples and 10 bones. It takes him 2 minutes to collect an apple and 1 minute to collect a bone. Carlos can reach his goal by first finding 7 bones, then trading trade 1 bone for 20 apples. This takes 7 minutes in total.

It can be shown that Carlos cannot reach his goal in less than 7 minutes.

§5 Pokémon Permutation

The insect Stentorceps weedlei *was named after the hairy bug Pokémon Weedle.*

Ash ran into a Pokémon on his way to Viridian City. Missing all of his Poké Balls, he had no way to capture the creature. Only the sounds it made before it escaped were recorded. Normally, Ash would be able to recognize the Pokémon by its appearance, but this species didn't appear in his Pokédex. Given its sounds, Ash hopes to identify the unknown species in the research lab at Viridian City. Unfortunately, his notes got all scrambled up in his rush to get to the lab. Specifically, he only has a permutation (a reordering) of the original string that he recorded.

As with most Pokémon, Ash knows that the only sound the unknown creature could have made was a repetition of its name two or more times. For example, if he had seen a Ditto, Ash could have recorded `dittoditto` or `dittodittoditto`. Can you help Ash identify the Pokémon by finding one of the possibilities of the original periodic string? It is possible that his notes may be wrong – if no answer exists, tell him so.

Input Format:

The first and only line contains a string of lowercase English letters. The length of the string is between 1 and 10^5 characters, inclusive.

Output Format:

Output a periodic permutation of the string. If no such string exists, output `IMPOSSIBLE`.

If there are multiple correct answers, you may print any.

Sample Input:

```
atatatraartttaarttata
```

Sample Output:

```
rattatarattatarattata
```

A valid possibility for the original periodic string is `rattatarattatarattata`. Note that this is not the only answer, for example `aaatttraaatttraaatttr` would also work.

§6 Island Isolation

Poptropica *was created by Jeff Kinney, the author of the* Diary of a Wimpy Kid *series.*

The *Poptropica* world consists of N islands and $N-1$ connections between pairs of islands. The connection between each pair of islands u and v consists of two directed bridges: one from u to v, and another from v to u (so there are $2\cdot(N-1)$ bridges in total). It is possible to reach any island from any other island by taking some sequence of bridges.

The evil genie Scheherazade wants to destroy all of Poptropica's bridges to generate more revenue for her blimp business. To prevent news of her evil ploy from spreading, the order in which she destroys the bridges is important. Specifically, before she destroys the directed bridge from u to v, she must make sure there is no way for people on island v to escape. That is, there may be no remaining bridges from v to another island w, besides possibly the bridge from v back to u.

Can you find an order in which Scheherazade can destroy the bridges? It can be proven that an answer always exists.

Input Format:

The first line contains N ($2 \leq N \leq 10^5$).

The next $N-1$ lines each contain two integers u_i and v_i, denoting the existence of a bridge from u_i to v_i and a bridge from v_i to u_i ($1 \leq u_i, v_i \leq N$; $u_i \neq v_i$).

Output Format:

Output $2\cdot(N-1)$ lines describing an order in which Scheherazade can remove the bridges. On each line, output two integers describing the starting and ending islands of the bridge to be destroyed.

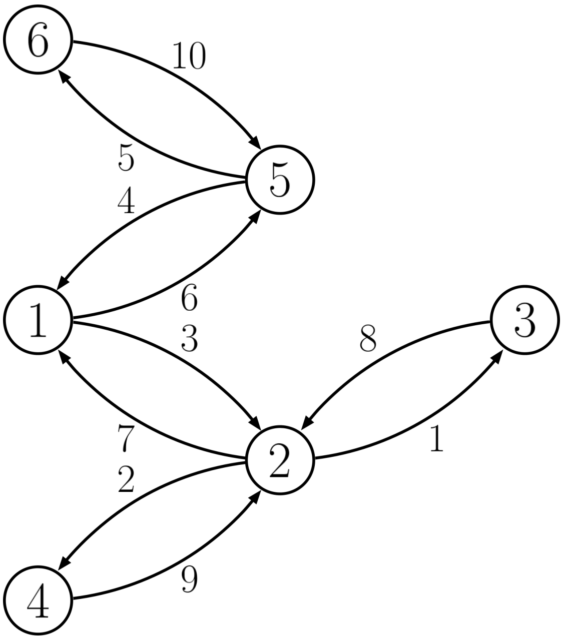
Sample Input:

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

Sample Output:

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

The order in which bridges are removed is shown here:



Note that this is not the only valid answer.

§7 Map Matching

Fortnite's *famous Orange Justice dance was popularized by a video of a kid wearing an orange shirt dancing.*

Leonel and his buddy Noel are playing *Fortnite* together. Unfortunately, they think that they might have dropped out of the Battle Bus onto separate island arenas! Leonel looks at his map and sees that his island is in the shape of a simple polygon with N vertices. When he plots these vertices on a coordinate plane, he gets the points $p_1, \ldots, p_N$. Noel also sees a simple polygonal island with N vertices on his map, but when he plots his points he gets $q_1, \ldots, q_N$.

Leonel and Noel want to know if their two polygons could describe the same island. The two friends might be in different positions and have their maps on different magnification levels. Formally, they want to know if one of their polygons can be scaled and translated to coincide exactly onto the other. Note that both friends are facing north, so the polygons cannot be rotated in this process.

To make matters worse, Leonel and Noel may have listed their points with different starting vertices (so p_1 doesn't necessarily correspond to q_1), although both sequences of points are guaranteed to be in clockwise order around the island. Can you figure out if their two polygons could describe same island shape?

Your program will have to answer multiple test cases.

Input Format:

The first line contains T, the number of test cases ($1 \le T \le 1000$).

The first line of each test case contains N ($3 \le N \le 10^5$). It is guaranteed that the sum of N over all T test cases does not exceed 10^5.

The next N lines of each test case describe the x and y coordinates of $p_1, \ldots, p_N$. The final N lines of each test case describe the x and y coordinates of $q_1, \ldots, q_N$. Both sequences of points are in clockwise order, and no three adjacent vertices in either polygon are collinear. All coordinates lie between -10^9 and 10^9, inclusive.

Output Format:

Output T lines, each containing `YES` or `NO`.

Sample Input:

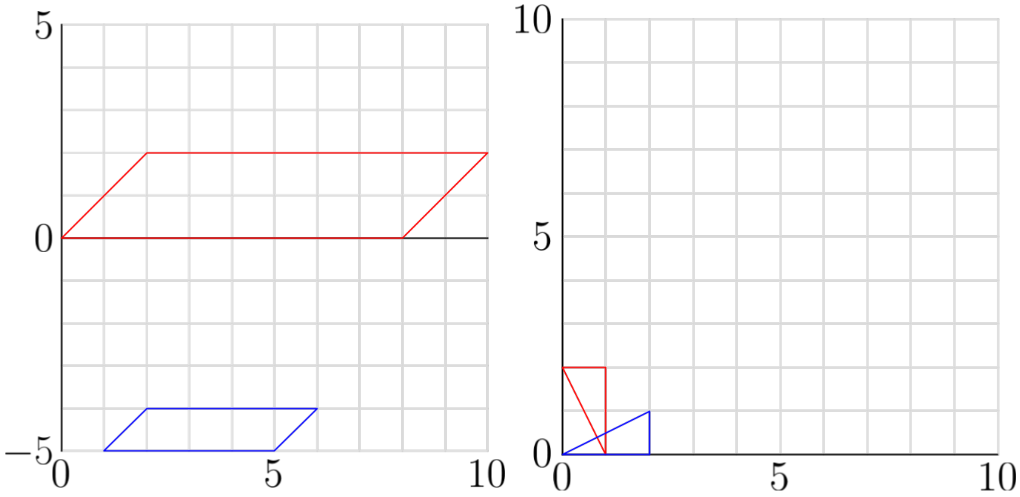
```
2
4
0 0
2 2
10 2
8 0
5 -5
1 -5
```

```
2 -4
6 -4
3
1 0
0 2
1 2
0 0
2 1
2 0
```

Sample Output:

```
YES
NO
```

The two polygons in each test case are shown here:



During the contest, there was a mistake in the diagram for the first case. We have fixed it here and apologize for the error.

In the first case, one polygon can be scaled and translated onto the other. In the second case, this is impossible. Note that rotations and flips are not allowed.

§8 Street Layout

There are a total of 24 SimCity *games, including versions for different gaming platforms, various sequels, and spin-offs.*

Beatriz is designing a mixed commercial and residential street in *SimCity*. The street has L equally-spaced plots of land where structures can be built, numbered $1, \ldots, L$. Currently, N of these plots are occupied by fire stations at the positions $a_1, \ldots, a_N$. Beatriz wants to select M of the remaining plots to build houses. The *risk* of a house is its distance to the nearest fire station, and the *overall risk* of a street layout is the maximum risk over all M houses. If Beatriz chooses a configuration of houses optimally, what is the minimum possible overall risk?

Input Format:

The first line contains N, M, and L ($1 \le N \le 10^5$; $1 \le M < L \le 10^9$; $N + M \le L$).

The next line contains N distinct integers $a_1, \ldots, a_N$ ($1 \le a_i \le L$). They may appear in any order.

Output Format:

Output one line with the minimum possible overall risk.

Sample Input:

```
2 3 7
4 6
```

Sample Output:

```
1
```

Let blanks be empty plots and F represent the locations of fire stations.

```
_ _ _ F _ F _
```

We choose to build houses at positions 3, 5, and 7.

```
_ _ H F H F H
```

Each house is 1 unit away from the nearest fire station, so the answer is 1.

§9 Grid Shuffling

2048 *was created by 19-year-old Gabriele Cirulli right after he graduated from high school!*

To hone her skills at the *2048* game, Natalia is shuffling tiles around an $N \times N$ grid. Initially, each cell of the grid is either empty or contains a colored tile. In a single move, she chooses a side of the grid (Up, Down, Left, or Right) and moves all of the tiles as far as she can in that direction. The tiles slide as they do in *2048*, except no two tiles ever merge, no matter their color. This means that the number of tiles never changes after a move.

To train her fingers, Natalia will execute the moves in a cyclical pattern: Down, Right, Up, Left, Down, Right, Up, Left, and so on. Can you determine the state of the grid after she finishes K total moves in this manner?

Input Format:

The first line contains N and K ($1 \leq N \leq 500$; $1 \leq K \leq 10^9$).

The next N lines each contain N integers describing the colors of the tiles in the grid. All colors are between 1 and N^2, inclusive. The colors are not necessarily distinct. Empty cells are represented with 0s.

Output Format:

Output N lines containing N integers each, describing the state of the grid after K moves.

Sample Input:

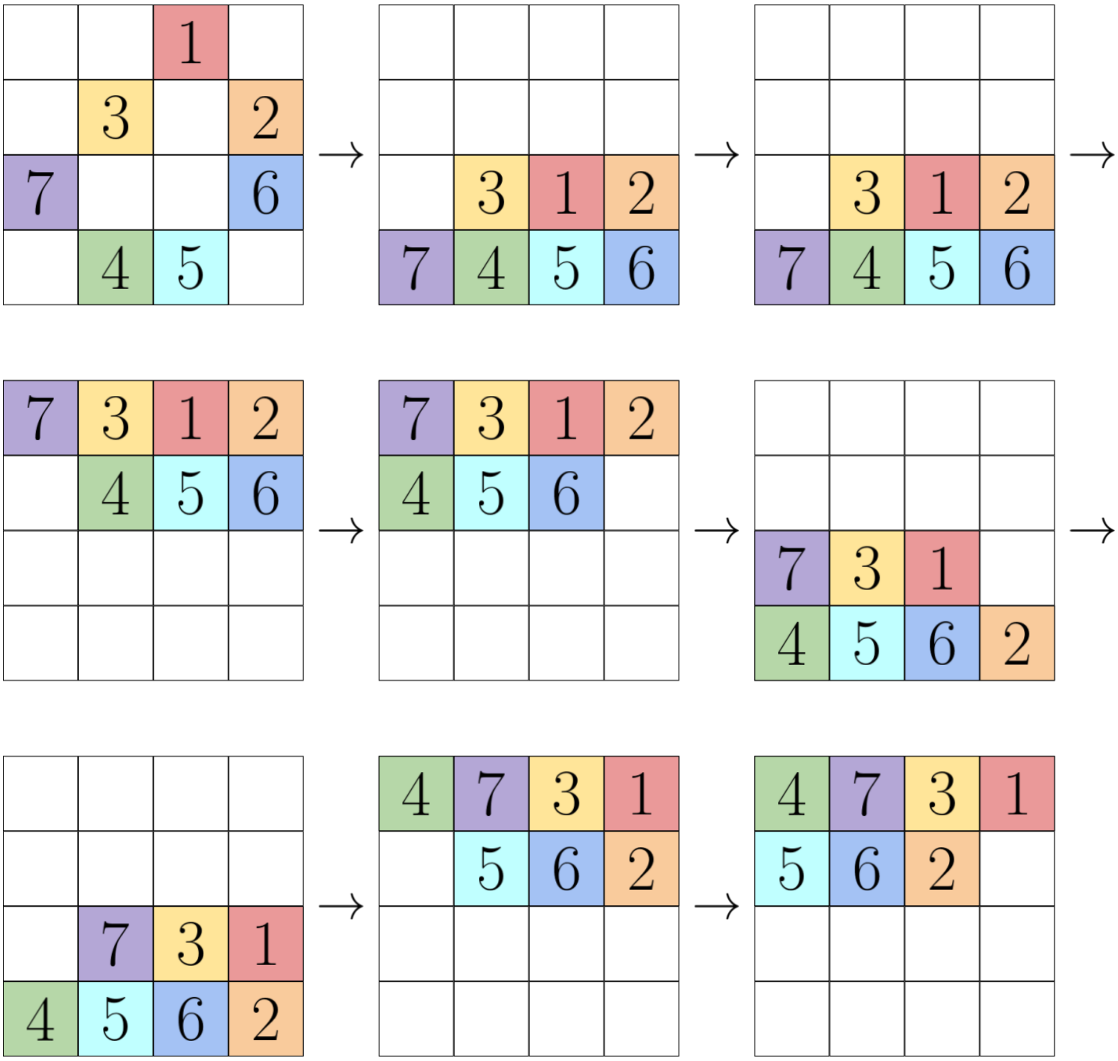
```
4 8
0 0 1 0
0 3 0 2
7 0 0 6
0 4 5 0
```

Sample Output:

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

See the diagram on the following page.

The positions of the tiles after each move are shown here:



		1	
	3		2
7			6
	4	5	

→

	3	1	2
7	4	5	6

→

	3	1	2
7	4	5	6

→

7	3	1	2
	4	5	6

→

7	3	1	2
4	5	6	

→

7	3	1	
4	5	6	2

→

	7	3	1
4	5	6	2

→

4	7	3	1
	5	6	2

→

4	7	3	1
5	6	2	

§10 Goomba Grouping

The famous character Mario first appeared in Donkey Kong, *where he had to rescue his girlfriend Pauline. Since then, he has appeared in more than 200 different video games.*

Bowser is coming up with a plan to stop his archenemy Mario. He commands an army of N Goombas of varying weights. To guard both entrances of his castle, he will split his army into two squadrons (each Goomba must be in exactly one squadron). Since bigger always means better, the strength of a squadron is equal to the sum of the weights of its Goombas. Bowser wants his two squadron strengths to be as close as possible to present a balanced defense.

Bowser comes up with the following greedy algorithm to split up his army. He first initializes two empty squadrons. He then iterates through his Goombas one by one in **decreasing** order of weight, each time adding it to the squadron that currently has the smaller total strength.

Bowser Jr. knows that his dad's solution will not always achieve the optimal splitting. To prove his dad wrong, Bowser Jr. wants to come up with a set of N Goomba weights for which an optimal splitting has squadrons with a strength difference of K, but his dad's algorithm results in a difference of more than K. Can you help Bowser Jr. construct such a set?

Your program will have to answer multiple test cases.

Input Format:

The first line contains T, the number of test cases ($1 \le T \le 100$).

The only line of each test case contains N and K ($1 \le N \le 20$; $0 \le K \le 10^9$). It is guaranteed that the sum of N over all T test cases will not exceed 100.

Output Format:

Output T lines. For each test case, if an answer exists, output N integers. Each integer must be between 1 and 10^{18}, inclusive. If no answer exists, output `-1`.

If there are multiple correct answers, you may print any.

Sample Input:

```
2
8 1
1 10
```

Sample Output:

```
9 13 16 15 19 20 15 18
-1
```

For the first test case, Bowser will take the following steps to build his squadron:

```
A = [], B = []
A = [20], B = []
A = [20], B = [19]
A = [20], B = [19, 18]
A = [20, 16], B = [19, 18]
A = [20, 16, 15], B = [19, 18]
A = [20, 16, 15], B = [19, 18, 15]
A = [20, 16, 15, 13], B = [19, 18, 15]
A = [20, 16, 15, 13], B = [19, 18, 15, 9]
```

The strength of squadron A is 64 and the strength of squadron B is 61, so Bowser's difference is 3. However, we can achieve a better answer of 1 with the following splitting:

```
A = [20, 15, 15, 13], B = [19, 18, 16, 9]
```

The strength of squadron A is 63 and the strength of squadron B is 62, so the difference is 1, which is indeed K (and less than Bowser's difference of 3).

For the second test case, Bowser's algorithm always achieves the optimal difference for all possible sets of size 1.

§11 Rabbit Subtraction

Clash of Clans *is one of the highest earning mobile games of all time, grossing over $7 billion since its release.*

Bored of standing on a tower all day, Willy the *Clash of Clans* wizard decides to practice spells on his N magical rabbits. When lined up from left to right, the i-th rabbit initially has the number a_i written on it. Whenever Willie declares the phrase *Lepus subtrahit*, a curious thing happens all at once: the first and second rabbits merge together, the third and fourth rabbits merge together, the fifth and sixth rabbits merge together, and so on (if the number of rabbits is odd, the last rabbit is left alone).

Upon merging, two rabbits with the numbers x and y fuse into a single rabbit with the number $x - y$ (x and y refer to the left and right rabbits, respectively). For example, if Willy has rabbits with the numbers `[3, 5, 6, 1, 3]`, after casting *Lepus subtrahit* he will be left with the rabbits `[-2, 5, 3]`. Note that the number of rabbits halves with each casting (technically, it goes from r to $\lceil \frac{r}{2} \rceil$).

Willy plans to use his spell over and over until there is only one rabbit left. His score will then be the number written on the final rabbit. Before he starts, he is allowed to move the first k rabbits to the end of the row, for some $0 \leq k < N$ (this results in a rotation of the original sequence). If Willy chooses k optimally, what is his maximum possible score?

Input Format:

The first line contains an integer N ($1 \leq N \leq 10^5$).

The next line contains N integers $a_1, \ldots, a_N$ ($0 \leq a_i \leq 10^9$).

Output Format:

Output one line with the Willy's maximum possible score.

Sample Input:

```
5
9 12 1 3 7
```

Sample Output:

```
6
```

If Willy chooses $k = 1$, his sequence of rabbits will start as `[12, 1, 3, 7, 9]`. As he performs his magic the sequence will become `[11, -4, 9]`, then `[15, 9]`, and finally `[6]`. It can be shown that 6 is Willy's highest possible score.

§12 Squid Art

Callie and Marie, the squid sisters in Splatoon, *are named after the squid dish calamari.*

Callie is playing a game of *Splatoon* on an $N \times M$ grid of hexagons, where each cell is either red or blue. She wants to paint the entire grid blue using a special bucket of paint. With a single use of said bucket, she can turn any contiguous component of blue hexagons red, or any contiguous component of red hexagons blue (this is a similar behavior to the paint bucket tool in most image editing programs). What is the minimum number of times Callie has to use the bucket to turn the **entire grid blue**?

Input Format:

The first line contains N and M $(1 \leq N \cdot M \leq 10^5)$.

The next N lines each contain a binary string of length M. Red and blue cells are denoted with 0s and 1s, respectively. The hexagonal grid is arranged such that the first cell in the top row is adjacent to the first two cells in the second row (see the diagram).

Output Format:

Output one line with the minimum number of times Callie has to use the paint bucket to turn the entire grid blue.

Sample Input:

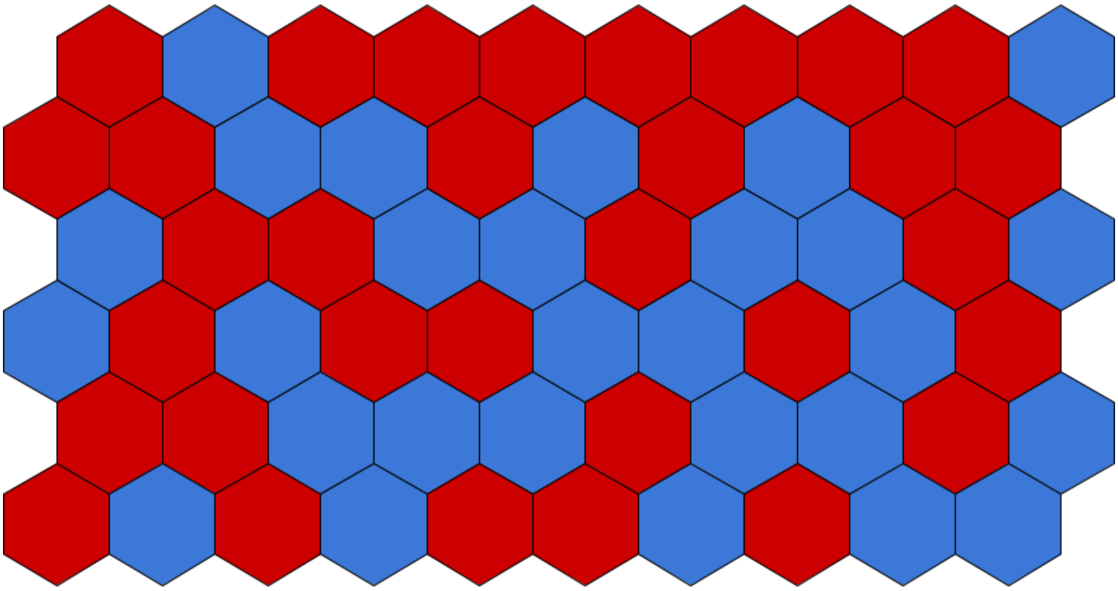
```
6 10
0100000001
0011010100
1001101101
1010011010
0011101101
0101001011
```

Sample Output:

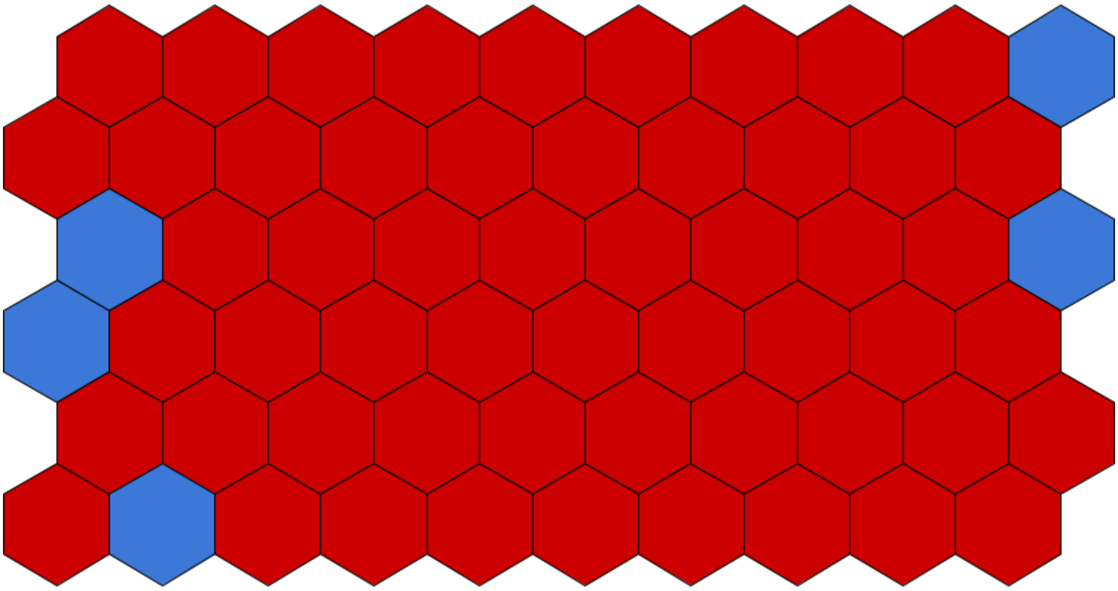
```
2
```

See the diagrams on the following page.

The initial coloring of the grid is shown here:



Callie can first paint the large blue component red. After she does so, the grid will look like this:



She can then paint the remaining red component blue. This process will turn every hexagon blue in 2 moves, which we can show is optimal.