Code Jam 2022 - Round 1C

Letter Blocks

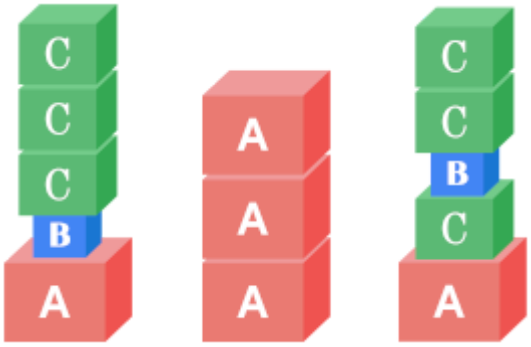
Problem

It is a rainy day, so you are indoors building towers of letter blocks. A letter block is a wooden cube that has a letter printed on one of its sides. The font used for the letters makes the blocks have a clear orientation: that is, there is only one side that can be pointed down (toward the floor) and one side that can be pointed up (toward the ceiling).

You have built multiple separate towers so far. Now you want to combine all of them into a single megatower by choosing one of your towers as the base, then picking up another tower (without changing the order of its blocks) and stacking the whole thing on top of that, and so on, until all towers have been used.

As an additional constraint for the megatower, for any two blocks that have the same letter, all blocks between them must also have that letter. That is, each letter of the alphabet that appears in the megatower needs to appear in one contiguous group (of one or more blocks).

For example, consider the following three possible megatowers. (These are separate examples, not built from the same original towers. Also note that the different block sizes are just for fun and are not part of the problem.)



The leftmost two megatowers are valid, since each letter appears in a contiguous group. However, the rightmost megatower is not valid, because there is a `B` in between two `C`s.

Given the towers that you have built so far, can you stack them all up into a valid megatower?

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow. Each test case is described by two lines. The first line consists of a single integer $\mathbf{N}$, the number of towers that are currently built. The second line consists of $\mathbf{N}$ strings $\mathbf{S_1}, \mathbf{S_2}, \ldots, \mathbf{S_N}$ representing the towers. Each of these strings consists of only uppercase letters. The i-th letter of each of these strings is the letter on the i-th block from the bottom in the represented tower.

Output

For each test case, output one line containing `Case #`x`:` y, where x is the test case number (starting from 1) and y is a string representing a valid megatower as described above, or the

word IMPOSSIBLE if no valid megatower can be built. (Notice that the string IMPOSSIBLE can never itself represent a valid megatower, since the two Is have other letters in between.)

Limits

Time limit: 5 seconds.
Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 100$.
$1 \leq$ the length of $\mathbf{S_i} \leq 10$, for all i.

Test Set 1 (Visible Verdict)

$2 \leq \mathbf{N} \leq 6$.

Test Set 2 (Visible Verdict)

$2 \leq \mathbf{N} \leq 100$.

Sample

Sample Input

```
6
5
CODE JAM MIC EEL ZZZZZ
6
CODE JAM MIC EEL ZZZZZ EEK
2
OY YO
2
HASH CODE
6
A AA BB A BA BB
2
CAT TAX
```

Sample Output

```
Case #1: ZZZZZJAMMICCODEEEL
Case #2: IMPOSSIBLE
Case #3: IMPOSSIBLE
Case #4: IMPOSSIBLE
Case #5: BBBBBAAAAA
Case #6: IMPOSSIBLE
```

In Sample Case #1, JAMMICCODEEELZZZZZ and ZZZZZJAMMICCODEEEL are the only two valid outputs.

In Sample Case #2, recall that *all* towers must be used in the megatower, so even though the first five towers together would form a valid megatower (as in Sample Case #1), the additional EEK makes the case impossible. No matter how the EEL and EEK towers are stacked relative to each other, there will be at least two non-contiguous groups of Es.

In Sample Case #3, no matter how you stack the towers, either the two Os are not contiguous or the two Ys are not contiguous.

In Sample Case #4, there are non-H letters in between the Hs of HASH, so this case is also impossible.

In Sample Case #5, this answer is the only valid one. Also notice that the towers are not necessarily all distinct.

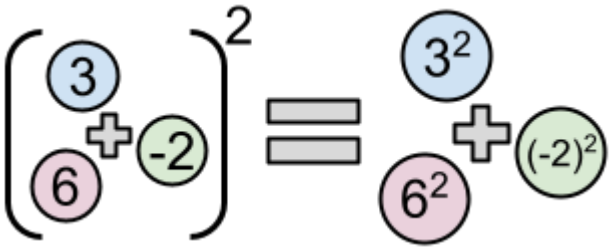
In Sample Case #6, no matter how you stack the towers, the two As cannot be contiguous.

Code Jam 2022 - Round 1C

Squary

Problem

Addition and squaring do not commute. That is, the square of the sum of all elements of a list of integers is not necessarily equal to the sum of the squares of those same elements. However, this is true for some lists; one example is $[3, -2, 6]$, because $(3 + (-2) + 6)^2 = 49 = 3^2 + (-2)^2 + 6^2$. Let us call these lists *squary*.



Given a (not necessarily squary) list of relatively small integers, we want to know whether it is possible to add at least 1 and at most $\mathbf{K}$ more elements such that the final list is squary. Each added element must be an integer between -10^{18} and 10^{18}, inclusive, and these do not have to be distinct from each other or from the initial list's elements.

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow. Each test case is described in two lines. The first line contains two integers $\mathbf{N}$ and $\mathbf{K}$, the number of elements of the initial list and the maximum number of elements you may add, respectively. The second line contains $\mathbf{N}$ integers $\mathbf{E_1}, \mathbf{E_2}, \ldots, \mathbf{E_N}$, representing the $\mathbf{N}$ elements of the initial list.

Output

For each test case, output one line containing `Case #`x`:` y, where x is the test case number (starting from 1). If it is possible to add at least 1 and at most $\mathbf{K}$ elements (each an integer between -10^{18} and 10^{18}, inclusive) to the initial list such that the square of the sum of its elements equals the sum of the squares of its elements, y should be $z_1\ z_2\ \ldots\ z_r$, where $1 \leq r \leq \mathbf{K}$ and the z_i values are the additional elements. If there is no way to accomplish this, y should be `IMPOSSIBLE`.

Limits

Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 100$.
$1 \leq \mathbf{N} \leq 1000$.
$-1000 \leq \mathbf{E_i} \leq 1000$, for all i.

Test Set 1 (Visible Verdict)

Time limit: 5 seconds.
$\mathbf{K} = 1$.

Test Set 2 (Visible Verdict)

Time limit: 10 seconds.
$2 \leq \mathbf{K} \leq 1000$.

Sample

Note: there are additional samples that are not run on submissions down below.

Sample Input

```
4
2 1
-2 6
2 1
-10 10
1 1
0
3 1
2 -2 2
```

Sample Output

```
Case #1: 3
Case #2: IMPOSSIBLE
Case #3: -10000000000000000000
Case #4: 2
```

In Sample Case #1, we can end up with the example list given in the problem statement.

In Sample Case #2, we have to add exactly one element. If we call that element x, the sum of the entire list is x and its square is x^2. The sum of the squares of all elements, on the other hand, is $x^2 + 10^2 + (-10)^2 = x^2 + 200 \neq x^2$, so the case is impossible.

In Sample Case #3, any integer in the $[-10^{18}, 10^{18}]$ range is a valid answer.

In Sample Case #4, notice that the input might contain duplicate elements, and that it is valid to create even more duplicates with the elements you choose to add.

Additional Sample - Test Set 2

The following additional sample fits the limits of Test Set 2. It will not be run against your submitted solutions.

Sample Input

```
3
3 10
-2 3 6
6 2
-2 2 1 -2 4 -1
1 12
-5
```

Sample Output

```
Case #1: 0
Case #2: -1 15
Case #3: 1 1 1 1 1 1 1 1 1 1 1
```

In Case #1 of the additional samples, we are given the example list from the problem statement, which is already squary, but we need to add at least one element to it. Adding a 0 keeps the list squary.

In Case #3 of the additional samples, we present one of multiple possible valid answers. Notice that it is permissible to add fewer than $\mathbf{K}$ elements; here $\mathbf{K}$ is 12 but we have only added 11 elements.

Code Jam 2022 - Round 1C

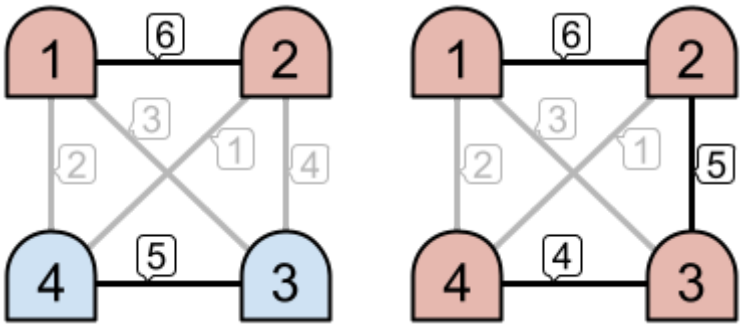
Intranets

Problem

Apricot Rules LLC is developing a new simplified networking protocol and wants to show off their routing algorithm. In their design, a network consists of $\mathbf{M}$ machines numbered from 1 to $\mathbf{M}$, and each pair of machines is connected by a direct link. Each of the links is given a unique integer priority value between 1 and $(\mathbf{M} \times (\mathbf{M}-1)/2)$ and each machine routes traffic according to those priorities.

Unfortunately, the routing algorithm is too aggressive and will route all traffic from a machine through the highest priority link connected to it. This may make some groups of machines isolated from others.

Formally, we say that a machine m uses a link ℓ if (and only if) ℓ is the highest priority link connected to m. We also say that a link is *active* if it is used by at least one of the two machines it connects. Given the link priorities, the original network becomes partitioned into disjoint intranets. Two machines belong to the same intranet if and only if there is some path between them using only active links.



For example, as seen in the left image above, only the links with priorities 6 and 5 are active. This creates two disjoint intranets. However, in the example on the right, three links are active, which results in one intranet consisting of all 4 machines.

As part of the quality assurance team at Apricot Rules LLC, you are investigating the extent of the problem. You are interested in knowing the probability of there being exactly $\mathbf{K}$ intranets if the priorities are assigned uniformly at random from among the $(\mathbf{M} \times (\mathbf{M}-1)/2)!$ ways of doing so.

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow. Each test case is described in a single line containing two integers $\mathbf{M}$ and $\mathbf{K}$: the number of machines and the target number of intranets, respectively.

Output

For each test case, output one line containing `Case #`x`:` y, where x is the test case number (starting from 1) and y is the sought probability computed modulo the prime $10^9 + 7$ (1000000007), which is defined precisely as follows. Represent the probability as as an irreducible fraction p/q (with p and q being non-negative integers that minimize $p + q$). Then, y

must equal $p \cdot q^{-1} \bmod 10^9 + 7$, where q^{-1} is the modular multiplicative inverse of q with respect to the modulus $10^9 + 7$. It can be shown that under the constraints of this problem, such a number y always exists and is unique.

Limits

Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 50$.
$1 \leq \mathbf{K} \leq \mathbf{M}/2$.

Test Set 1 (Visible Verdict)

Time limit: 20 seconds.
$2 \leq \mathbf{M} \leq 50$.

Test Set 2 (Hidden Verdict)

Time limit: 60 seconds.
$2 \leq \mathbf{M} \leq 5 \times 10^5$.

Sample

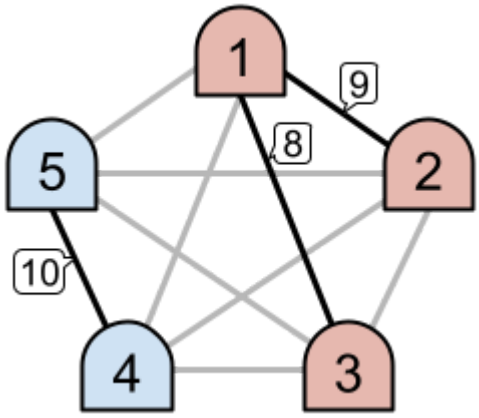
Sample Input

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

Sample Output

```
Case #1: 428571432
Case #2: 571428576
Case #3: 47619048
```

In Sample Case #1, consider the following situation. Let's call $\mathbf{M} = 5$ machines $1, 2, 3, 4, 5$ and denote the link connecting machine a and machine b by (a, b). Assume that the priorities of links $(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)$ are $9, 8, 7, 6, 5, 4, 3, 2, 1, 10$, respectively. Then machines 1 and 2 use link $(1, 2)$, machine 3 uses link $(1, 3)$, and machines 4 and 5 use link $(4, 5)$. Thus three links $(1, 2), (1, 3), (4, 5)$ are active, and there are two intranets $\{1, 2, 3\}$ and $\{4, 5\}$. Since $\mathbf{K} = 2$, this situation counts the answer.



We can find that there are 1555200 ways to assign the priorities to have exactly 2 intranets among $10! = 3628800$ ways, so the probability is $3/7$.

In Sample Case #2, the probability is $4/7$.

In Sample Case #3, the probability is $1/21$.