A – Three Castles

Memory limit: 1024 MB
Time limit: 4 s

EUC 2026
8.02.2026

In the distant kingdom of ICPC, there are n towers. For each tower we know its position (x, y) on the kingdom's map. For simplicity, we treat each tower as a point on the map. It is guaranteed that there are no two towers in the same position, and no three towers lie on the same line.

King EUC has decided that it is time to build three castles, according to the following rules:

- Each castle contains a non-empty set of towers.
- No tower should be left alone, not belonging to any castle.
- Every castle should form a convex polygon on the kingdom's map with all its towers located in the vertices of this polygon. We assume that castles with one or two towers are also valid.
- No pair of castles should have a point in common, be it at the border or in the interior.

King EUC has not yet decided the number of towers for each of its three castles. Therefore, he asks you to determine for every triplet of positive integers s_1, s_2, s_3 (such that $s_1 + s_2 + s_3 = n$), if there is a solution where the number of towers in the i-th castle is s_i $(1 \leq i \leq 3)$, and if such a solution exists, to show an example of a partition of towers among three castles.

Input

The first line of the input contains one integer n $(3 \leq n \leq 40)$, denoting the number of towers. Towers are numbered from 1 to n.

Each of the next n lines contains the coordinates of one tower, two integers x and y $(-10^6 \leq x, y \leq 10^6)$. As it was already mentioned, there are no two towers in the same position, and no three towers lie on the same line.

Output

The first line of the output should hold the number k, being the number of different triplets for which a solution exists.

Each of the next k lines should contain a solution to one of the triplets. Each line should consist of $3 + n$ integers separated by single spaces: s_1, s_2, s_3, $a_{1,1}, \ldots, a_{1,s_1}$, $a_{2,1}, \ldots, a_{2,s_2}$, $a_{3,1}, \ldots, a_{3,s_3}$, where $a_{i,j}$ denotes the number of the j-th tower in the i-th castle.

Each unordered triplet with a solution should appear exactly once, and the order of the towers within each castle can be arbitrary.

A – Three Castles

Memory limit: 1024 MB
Time limit: 4 s

EUC 2026
8.02.2026

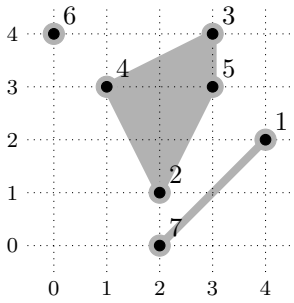
Example

For the input data:

```
7
4 2
2 1
3 4
1 3
3 3
0 4
2 0
```

one possible correct result is:

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



Explanation: There are four possible triplets that sum to 7. For triplet $1, 2, 4$ one solution is to take castles $\{6\}$, $\{1, 7\}$, and $\{2, 3, 4, 5\}$. Triplets $1, 3, 3$ and $2, 2, 3$ also have solutions. There is no solution for triplet $1, 1, 5$.

Note that $\{4\}$, $\{2, 5\}$, and $\{1, 3, 6, 7\}$ is not a valid solution for triplet $1, 2, 4$, since the third castle intersects with the first and the second. Note that $\{2\}$, $\{7\}$, and $\{1, 3, 4, 5, 6\}$ is not a valid solution for triplet $1, 1, 5$, since the third castle is not convex.

B – Bitwise Beach

Memory limit: 1024 MB
Time limit: 1 s

EUC 2026
8.02.2026

Two friends collected n^2 shells at a beach, and arranged them in a grid of n rows and n columns. They then drew some horizontal and some vertical lines in the sand, which partitioned the beach into zones, with a positive number of shells from their collection in each zone.

Now, the friends play a well-known game of Nim. They make alternating moves; in each move, a player takes a positive number of shells from any single zone. The player who takes the last shells wins.

The friends know that to determine whether the first player has a winning strategy, and which move is optimal, one should count the number of shells in every zone and calculate the bitwise xor of these numbers. That sounds like a lot of work. Help them!

Input

The first line of the input contains one integer n ($1 \le n \le 10^6$). We assume that the rows are numbered 1 through n, top to bottom. The columns are also numbered 1 through n, left to right.

The second line contains a string of $n-1$ digits 0 or 1; the i-th digit ($1 \le i < n$) is 1 if there is a horizontal line between row i and row $i+1$.

The third line contains a string of $n-1$ digits 0 or 1; the i-th digit ($1 \le i < n$) is 1 if there is a vertical line between column i and column $i+1$.

Output

Output a single integer, the bitwise xor of the numbers of shells in all zones.

Example

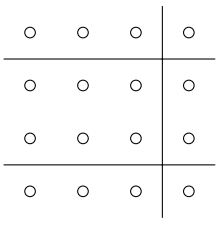
For the input data:

```
4
001
101
```

the correct result is:

```
4
```

Explanation: There are six zones, having 3, 1, 6, 2, 3, 1 shells, respectively. The bitwise xor is equal to $3 \oplus 1 \oplus 6 \oplus 2 \oplus 3 \oplus 1 = 4$.



C – Copy-paste

Memory limit: 1024 MB
Time limit: 1 s

EUC 2026
8.02.2026

John takes security very seriously, so his password is very long: it has n characters. However, he is not very creative, so all characters in the password are the same—the letter a. John is very impatient, so he would like to type the password as fast as possible.

He wants to obtain a string consisting of exactly n letters a. He can use a sequence of the following actions: "a" (type the letter a), "Ctrl-A" (select everything), "Ctrl-C" (copy to the clipboard), "Ctrl-V" (paste from the clipboard). What is the minimal number of actions needed to achieve the desired result when starting from an empty password box and an empty clipboard?

Below are precise definitions of all four actions John can take in the password box:

- Pressing "a": If no text is selected, insert the letter a at the position of the cursor (to the left). Otherwise, replace the selected text with the letter a. After that, no text is selected and the cursor is positioned at the end of the password box.
- Pressing "Ctrl-A": Select all text in the password box.
- Pressing "Ctrl-C": If no text is selected, do nothing. Otherwise, copy the selected text to the clipboard.
- Pressing "Ctrl-V": If the clipboard is empty, do nothing. Otherwise, if no text is selected, insert the text from the clipboard at the position of the cursor (to the left). Otherwise, replace the selected text with the text from the clipboard. In the end, no text is selected and the cursor is positioned at the end of the password box. The contents of the clipboard do not change.

Input

The first and only line of the input contains one integer n ($1 \leq n \leq 10^{12}$), denoting the desired length of the password.

Output

Output one integer, the minimal number of actions needed to type the password with the desired length.

Example

For the input data:	the correct result is:
11	10

Explanation: John can use the following sequence of actions: a, a, Ctrl-A, Ctrl-C, Ctrl-V, Ctrl-V, Ctrl-V, Ctrl-V, Ctrl-V, a.

D – Deque Sort

Memory limit: 1024 MB
Time limit: 4 s

EUC 2026
8.02.2026

There are n parcels, numbered from 1 to n, that just arrived to a warehouse. They came in the order $a_1, \ldots, a_n$, and we want to sort them so that they are in the order $1, 2, \ldots, n$.

The sorting machine in the warehouse works in phases. Each phase accepts a row $a_1, \ldots, a_n$ of parcels and proceeds as follows: First, the first parcel a_1 is put into a new row. Then, for every parcel a_i for $i = 2, \ldots, n$, if a_i is greater than the last (rightmost) parcel in the new row, it puts it at the end of the row. Otherwise, it puts it at the front of the row.

If the new row is not sorted, it serves as the input for the next phase of the sorting machine.

Write a program that helps testing whether the machine works as intended by answering queries of the form "which parcel is at the b-th place after t phases".

Input

The first line of the input contains two integers n and q ($1 \leq n, q \leq 5 \cdot 10^5$), denoting the number of parcels and the number of queries, respectively.

The second line contains a sequence of n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$, $a_i \neq a_j$) that represent the initial ordering of the parcels.

The next q lines contain queries; the i-th line contains two integers t_i and b_i ($0 \leq t_i \leq 10^9$, $1 \leq b_i \leq n$), denoting the i-th query.

Output

Output exactly q lines. The i-th line should contain the answer to the i-th query.

If the machine performed less than t_i phases, write -1. Otherwise, write which parcel is at the position b_i from left, after t_i phases have been performed.

Example

For the input data:

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

the correct result is:

```
4
2
-1
5
```

Explanation: The machine will perform three phases:

$$2\,1\,3\,5\,4 \quad \to \quad 4\,1\,2\,3\,5 \quad \to \quad 3\,2\,1\,4\,5 \quad \to \quad 1\,2\,3\,4\,5.$$

Whereas for the input data:

```
1 1
1
1 1
```

the correct result is:

```
-1
```

E – Evaluation

Memory limit: 1024 MB
Time limit: 1 s

EUC 2026
8.02.2026

We have a string consisting of n non-zero digits. We can produce an expression by inserting at most one of the characters + (plus) or · (multiply) between any two consecutive digits. For instance, from the string "231" we can produce nine expressions:

$$231, \quad 23+1, \quad 23\cdot 1, \quad 2+31, \quad 2\cdot 31, \quad 2+3+1, \quad 2+3\cdot 1, \quad 2\cdot 3+1, \quad 2\cdot 3\cdot 1.$$

We evaluate these expressions with the standard order of operations, and calculate the sum of all values. For our example we get

$$231+24+23+33+62+6+5+7+6=397.$$

Calculate the above sum modulo 10^9+7.

Input

The first line of the input contains one integer n ($1 \le n \le 10^6$), denoting the length of the string. The second line contains a string of length n consisting of digits 1–9.

Output

Output one integer, the sum of all expressions modulo 10^9+7.

Example

For the input data:	the correct result is:
3 231	397

F – Fix the Coloring

Memory limit: 1024 MB
Time limit: 6 s

EUC 2026
8.02.2026

You are given an undirected graph of n vertices and m edges. Each vertex is colored either black or white. You would like the coloring to be proper. That is, no two vertices connected by an edge should have the same color.

It is possible that some pairs of vertices do not satisfy this property. You want to fix this by repainting some vertices. However, you have only red paint, so you will repaint some vertices red.

But even with three colors at hand fixing the coloring could be impossible. For example, when the graph contains a big clique (that is, a set of vertices in which all pairs are connected with edges). You decided to allow at most one nontrivial clique in this graph to have all its vertices painted red.

So, your task is to check whether it is possible to paint some vertices red in such a way that:

- All edges that connect red vertices form a clique.
- All remaining edges connect vertices of different colors.

Input

The first line of the input contains one integer t ($1 \le t \le 1000$) denoting the number of test cases.

Each test case starts with a line containing two integers n and m ($1 \le n, m \le 3000$), denoting the numbers of vertices and edges in the graph. Vertices are numbered from 1 to n.

The second line contains a string of n characters `B`, `W`; the i-th character denotes the color of the i-th vertex (`B` for black and `W` for white).

The next m lines contain the edges of the graph, each described by a pair of vertex numbers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$). Each unordered pair u, v will appear in the text case at most once.

The sum of values n and the sum of values m over all test cases do not exceed 3000.

Output

Output the answers to all t test cases.

If fixing the coloring is not possible, the answer is one line consisting of the word `NO`. Otherwise, the answer should consist of two lines: The first line should contain one word `YES`, and the second line a string of n characters `B`, `W`, `R` denoting the new colors of vertices (where `R` denotes red). If there are multiple valid solutions, you may output any one of them.

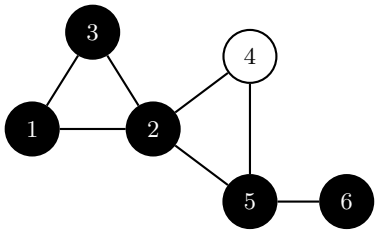
Example

For the input data:

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

a possible correct result is:

```
YES
RRRWBR
NO
```



Explanation: In the first test case, one solution is to repaint the clique 1, 2, 3 and vertex 6. Another solution is to repaint the clique 1, 3 and vertex 5.

G – Good Permutations

Memory limit: 1024 MB
Time limit: 3 s

EUC 2026
8.02.2026

You are given an array $a_1, \ldots, a_n$ of length n consisting of integers in $\{1, \ldots, n\} \cup \{-1\}$.

A sequence of integers $p_1, \ldots, p_k$ is called *a good permutation* if $\{p_1, \ldots, p_k\} = \{1, \ldots, k\}$ and $p_{i+1} \geq p_i - 1$ holds for all $1 \leq i < k$.

You are to answer q queries. Each query is specified by a pair of integers (l, r). For such a query, check if it is possible to replace all -1s in the subarray $a_l, \ldots, a_r$ with positive integers in such a way that this subarray becomes a good permutation.

Input

The first line of the input contains two integers n and q ($1 \leq n, q \leq 2 \cdot 10^5$), denoting the length of the array and number of queries, respectively.

The second line contains a sequence of n integers $a_1, a_2, \ldots, a_n$ ($-1 \leq a_i \leq n$, $a_i \neq 0$) denoting the elements of the array.

The next q lines describe queries; the i-th line contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq n$) representing the i-th query.

Output

Output exactly q lines. The i-th line should contain the answer to the i-th query, being a word `YES` or `NO`.

Example

For the input data:

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

the correct result is:

```
YES
NO
YES
```

Explanation: In the first query we can have a good permutation $3, 2, 1, 4, 5$. In the second query, no good permutation of 4 elements can contain the element 5. In the third query we can have a good permutation $2, 1, 3$.

Whereas for the input data:

```
2 2
1 1
1 2
1 1
```

the correct result is:

```
NO
YES
```

H – House

Memory limit: 1024 MB
Time limit: 2 s

EUC 2026
8.02.2026

Byteasar would like to build a wooden house. He has n pieces of wood of lengths $a_1, \ldots, a_n$ and equal heights. That wood will be used for the walls of the house.

Byteasar decided that the floor of the house will be a rectangle, and he would like to make it as big as possible. For a rectangle floor of size $x \times y$, the area of the house will be $x \cdot y$.

The house will have four walls, each being a rectangle. For each wall Byteasar will use exactly k planks that will be stacked on top of each other. Therefore he will need $2k$ planks of length x and $2k$ planks of length y.

Byteasar is free to cut the pieces of wood into planks however he likes. Thus from a single piece of wood of length a_i he can obtain any sequence of planks of real lengths $b_1, \ldots, b_m$ as long as $b_1 + \cdots + b_m \leq a_i$. Byteasar does not have to use all the wood; any leftover scraps may remain.

Help him and calculate the maximal area of the house he can obtain.

Input

The first line of the input contains two integers n and k ($1 \leq n \leq 1000$; $1 \leq k \leq 30$), denoting the number of pieces of wood and the height of the walls.

In the second line there is a sequence of n integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq 1000$), denoting the lengths of the pieces of wood.

Output

Write one real number, the maximal possible area of the house (that is, the maximal value of $x \cdot y$, provided that Byteasar can obtain $2k$ planks of length x and $2k$ planks of length y).

The allowable relative or absolute error is 10^{-9}. That is, if you output S and the correct exact result is R, then it must hold that $|S - R| \leq 10^{-9} \cdot \max(1, R)$. You may print at most 20 digits after the decimal point.

Example

For the input data:

```
1 5
10
```

a correct result is:

```
0.25
```

Whereas for the input data:

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

a correct result is:

```
12
```

And for the input data:

```
5 7
1 4 5 7 5
```

a correct result is:

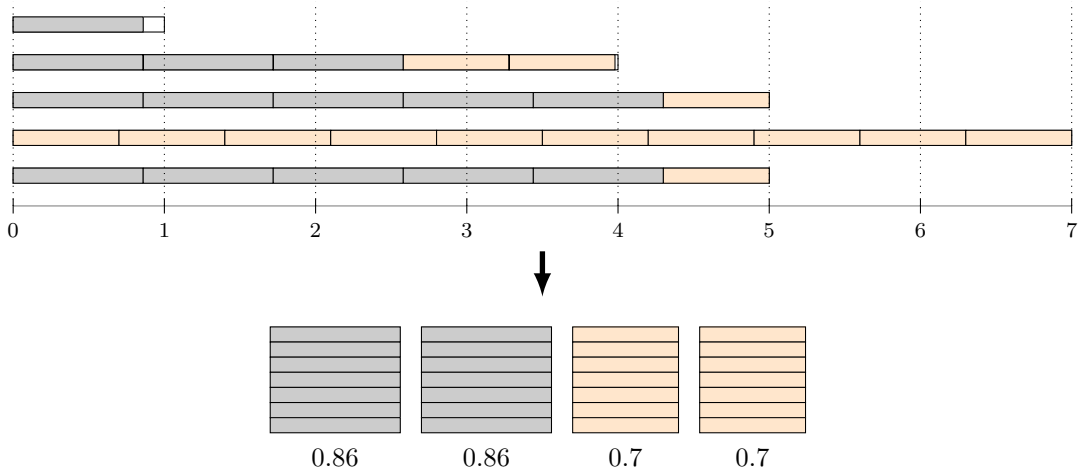
```
0.602
```

Explanation: In the first example we need 20 planks (5 for each wall). The best is to cut the only piece of wood into 20 equal parts of length 0.5 each, and build a house of area $0.5 \cdot 0.5 = 0.25$.

In the second example we have two distinct solutions that allow to obtain a house of area 12:

- For a house of size 6×2, we need to shorten a piece of length 7 to 6 and a piece of length 3 to 2.
- For a house of size 4×3, we need to cut a piece 7 into planks of length 3 and 4, and shorten a piece 6 to 4.

In the third example, the optimal house has dimensions 0.7×0.86. The picture below shows how to obtain 14 planks of length 0.86, 14 planks of length 0.7 and two scraps of length 0.14 and 0.02:



I – Palindromes

Memory limit: 1024 MB
Time limit: 3 s

EUC 2026
8.02.2026

In this problem we consider strings over the binary alphabet $\{\texttt{a}, \texttt{b}\}$. A string is called a *palindrome* if it reads the same from left to right and from right to left. For example, the strings `a`, `aba`, and `babbab` are palindromes, but `abab` is not.

There is an unknown binary string s of length N over the alphabet $\{\texttt{a}, \texttt{b}\}$. We only know that its prefixes (that is, initial fragments) of lengths $a_1, \ldots, a_n$ are palindromes. In how many ways can the string s be recovered? The answer can be quite large; it suffices to return its remainder modulo m.

Note that the sought string s may have other prefixes being palindromes, in addition to the ones of lengths $a_1, \ldots, a_n$.

Input

The first line contains three integers N, n, and m ($1 \leq N \leq 10^{18}$, $1 \leq n \leq 500\,000$, $2 \leq m \leq 10^9$) that represent the length of the string, the number of palindromic prefixes of the string, and the number used to compute the result.

The second line contains a strictly increasing sequence of n integers $a_1, a_2, \ldots, a_n$ (with $1 \leq a_i \leq N$) that denote the lengths of the palindromic prefixes of the sought string.

Output

The output should contain one non-negative integer, the number of strings of length N that have a palindromic prefix of each length $a_1, \ldots, a_n$, modulo m.

Example

For the input data:

```
10 3 100
2 5 9
```

the correct result is:

```
8
```

Explanation: There are 8 strings of length 10 whose prefixes of lengths 2, 5, and 9 are palindromes:

1. `aaaaaaaaaa`
2. `aaaaaaaaab`
3. `aabaaabaaa`
4. `aabaaabaab`
5. `bbabbbabba`
6. `bbabbbabbb`
7. `bbbbbbbbba`
8. `bbbbbbbbbb`

J – Jewel Guards

Memory limit: 1024 MB
Time limit: 2 s

EUC 2026
8.02.2026

We are organizing a jewel exhibition. Your task is to make sure that the jewels are safe at night.

There are k guards available. For each guard, it is known when the guard can work during the night. We assume that each night is divided into n time units. The jewels are considered *safe* if, during at least m time units of the night, they are watched by at least *two* guards.

To avoid the guards being bribed, you want to select different subsets of guards each night so that the jewels are safe during every night. Compute how many such subsets exist.

Input

The first line contains three integers n, k, and m ($1 \le n \le 10^9$, $2 \le k \le 20$, $1 \le m \le n$) that denote the number of time units of the night, the number of guards, and the number of time units of the night during which at least two guards must watch the jewels.

The next k lines describe time intervals during which each guard is available for work. Each line starts with a non-negative integer c_i ($i \in \{1, \ldots, k\}$) that denotes the number of time intervals. What follows are c_i pairs of integers $a_{i,j}$, $b_{i,j}$ ($j \in \{1, \ldots, c_i\}$) that describe intervals $[a_{i,j}, b_{i,j}]$ such that the i-th guard works from time unit $a_{i,j}$ to time unit $b_{i,j}$, inclusive. The intervals are pairwise disjoint and are listed in the order of increasing left endpoints.

You may assume that $c_1 + c_2 + \cdots + c_k \le 10^6$.

Output

The output should contain one non-negative integer, the number of subsets of guards that can be selected so that the jewels are safe during the night.

Example

For the input data:

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

the correct result is:

```
2
```

Explanation: The sought subsets of guards are $\{1, 3\}$ and $\{1, 2, 3\}$. The first and third guard both watch the jewels during six time units, $\{1, 2, 4, 8, 9, 10\}$. If all three guards are selected, the jewels are watched by at least two guards during eight time units $\{1, 2, 3, 4, 5, 8, 9, 10\}$.

K – Multiset Variance

Memory limit: 1024 MB
Time limit: 2 s

EUC 2026
8.02.2026

We have n multisets of integers. We create a nonempty multiset by taking each multiset from the input in whole arbitrarily many times (possibly 0 times).

For example, from input multisets $\{1,3\}, \{1,2,2,3\}, \{3,5\}$, we can create a multiset $M = \{1,3,1,3,1,2,2,3\}$ by taking the first one twice, the second one once, and not taking the third one.

What is the minimum and maximum variance of a multiset that we can achieve?

Recall that the mean $\mathrm{E}(X)$ and variance $\mathrm{Var}(X)$ of a multiset $X = \{x_1, x_2, \ldots, x_k\}$ are defined as follows:

$$\mathrm{E}(X) = \frac{1}{k}\sum_{i=1}^{k} x_i, \qquad \mathrm{Var}(X) = \frac{1}{k}\sum_{i=1}^{k} (x_i - \mathrm{E}(X))^2.$$

For instance, the mean and variance of the above multiset M are:

$$\mathrm{E}(M) = \tfrac{1}{8}(1+3+1+3+1+2+2+3) = \tfrac{16}{8} = 2,$$
$$\mathrm{Var}(M) = \tfrac{1}{8}((-1)^2+1^2+(-1)^2+1^2+(-1)^2+0^2+0^2+1^2) = \tfrac{6}{8} = \tfrac{3}{4}.$$

Input

The first line of the input contains one integer t ($1 \le t \le 1000$), denoting the number of test cases.

Each test case starts with a line containing one integer n ($1 \le n \le 10^5$), denoting the number of multisets.

The next n lines describe the multisets. Each line starts with an integer k ($1 \le k \le 10$) denoting the size of the multiset, followed by a sequence of k integers from the range $[-2 \cdot 10^5, 2 \cdot 10^5]$ representing the elements of the multiset.

The sum of the sizes of all multisets over all test cases does not exceed 10^6.

Output

Output exactly t lines containing answers to subsequent test cases.

The answer consists of two real numbers separated by a space, the minimum and maximum variance. The allowable relative or absolute error is 10^{-11}. That is, if you output S and the correct exact result is R, then it must hold that $|S - R| \le 10^{-11} \cdot \max(1, R)$. You may print at most 20 digits after the decimal point.

It can be shown that there is always a solution that achieves the minimum variance and a solution that achieves the maximum variance.

K – Multiset Variance

Memory limit: 1024 MB
Time limit: 2 s

Example

For the input data:

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

a correct result is:

```
0.5 2
2 2
0 2.777777777778
```

Explanation: In the first test case, the minimum variance (equal to $\frac{1}{2}$) is achieved for the multiset $\{1, 2, 2, 3\}$, and the maximum (equal to 2) for the multiset $\{1, 3, 3, 5\}$.

In the second test case, both minimum and maximum variance (equal to 2) is obtained for the multiset $\{-3, 0, 0, -2, -2, 1\}$.

In the third test case, the minimum variance (equal to 0) is achieved for the multiset $\{5\}$, and the maximum (equal to $\frac{25}{9}$) for the multiset $\{1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 5, 5, 5, 5, 5, 5, 5, 5\}$.