2018 LGE Code Jam: Final Round

Problem A. Movers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 sec
Memory limit:	512 MB

Alex and his movers are moving items from a house. This household has exactly A items each of which is 1 kg, B items 2kg, C items 3kg, D items 4 kg, and E items 5 kg.

In order to move them, items should be put into a box where each box can hold up to 5kg. Alex wants to minimize the number boxes his crew needs, while putting all items into boxes.

The sum of the weight of items in any box cannot exceed 5kg.

Given how many items of each weigh there are, compute the minimum number of boxes Alex needs.

Input

The first line contains five integers, A, B, C, D, and E ($0 \le A, B, C, D, E \le 1,000$).

Output

You must output in a single line the minimum number of boxes Alex needs.

standard input	standard output
5 0 0 0	1
0 0 0 0 5	5
1 1 1 1 1	3
10 8 2 7 3	15
8 2 7 3 0	10

Problem B. Roll cake

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 sec
Memory limit:	512 MB

It's Jaehyun's birthday today. Jaehyun got N roll cakes from N friends, one from each friend. Each roll cake's length is $A_1, A_2, \ldots, A_N$. Jaehyun only eats roll cakes of length exactly 10. Therefore, he wants to create as many roll cakes of length 10 as possible by cutting those N roll cakes.

A roll cake can be cut into smaller ones, according to the following procedure.

1. First, pick a roll cake to be cut – its length must be greater than 1. Let x be the length of the chosen roll cake.
2. Then, pick an integer y that is greater than 0 and less than x.
3. The roll cake is cut into two smaller ones of length y and $x - y$, respectively.

Jaehyun can cut roll cakes at most M times. Compute the maximum number of roll cakes of length 10 that he can get.

Input

The first line contains N, the number of roll cakes, and M, the maximum number of cuts Jaehyun can perform using the machine ($1 \leq N, M \leq 1,000$).

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 1,000$) where A_i describes the length of an i-th roll cake.

Output

Output the maximum number of roll cakes of length 10 that Jaehyun can make.

Examples

standard input	standard output
3 1 10 10 10	3
3 1 10 10 20	4
3 3 20 20 20	6
5 7 10 20 30 40 50	11
5 8 34 45 56 12 23	8

Problem C. Rectangles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 sec
Memory limit:	512 MB

Alex found N sticks that he played with when he was little. Sticks' lengths are $A_1, A_2, \ldots, A_N$, and the length is an integer that is greater than or equal to 2.

Alex wants to create rectangles using these sticks. Each stick can be used at most once, and he cannot use multiple sticks to form a side of the rectangle. Not all sticks need to be used. Alex can make more than one rectangle.

Alex also has a machine that can reduce the length of a stick by 1. If the original length of a stick was A_i, then it can be reduced to $A_i - 1$ using this machine. Alex can use the machine as many times as he wants to, but he cannot reduce the length of the same stick more than once.

Alex wants to make rectangles using these N sticks such that the sum of areas of the rectangles is maximized. You must compute this maximum area he can achieve.

Input

The first line contains N ($1 \leq N \leq 10,000$), the number of sticks.

The second line contains N integers describing the lengths of N sticks, $A_1, A_2, \ldots, A_N$ ($2 \leq A_i \leq 100,000$).

Output

Output the maximum total area of rectangles Alex can achieve.

Examples

standard input	standard output
4 5 5 6 6	30
4 4 5 2 3	8

Alex can make a rectangle with sides of length 2 and 4.

standard input	standard output
4 2 4 6 8	0
6 5 6 6 3 4 4	24
9 10 3 4 4 4 5 6 6 6	42

Using the last four sticks (after reducing one 6-length stick's length by 1), Alex can make a rectangle whose area is 5x6 = 30. Then, using the other four sticks of lengths 3, 4, 4, and 4, he can make another rectangle whose area is 3x4 = 12. There is no way to make rectangles whose sum of area is greater than 42.

standard input	standard output
10 10 10 10 10 10 10 10 10 10 10	200
4 100000 100000 100000 100000	10000000000

Problem D. Team Formation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 sec
Memory limit:	512 MB

Albert's school is hosting a hackathon event for students. There are N students who wish to participate in the event. Let's label the students as 1, 2, ..., N.

Student i wants to participate in the event if and only if the team he/she belongs to has at most X_i members in it (including student i). Because the space is limited at the event venue, the school needs to minimize the number of teams. More precisely, Albert wants to compute the minimum number of teams while meeting the following conditions:

- Every student is assigned to exactly one team
- Each team contains at least one student
- If student i belongs to a team, then the number of students in the team (including student i) is no greater than X_i.

Albert needs to help his school host the event successfully, and you need to help Albert determine the minimum number of teams that can be formed given students' constraints.

Input

The first line will contain the number of students, N ($1 \le N \le 100,000$).

The second line will contain N integers representing X_i values for each student. Assume that $1 \le X_i \le N$ for all i's.

Output

You should output a single number, representing the minimum number of teams that can be formed.

Examples

standard input	standard output
2 2 2	1
5 1 2 5 2 1	4
5 1 2 1 2 1	4
9 2 2 2 3 3 3 2 2 2	4

Here is one solution to assign students to four teams:

{1, 9}, {2, 8}, {3, 7}, {4, 5, 6}.

standard input	standard output
9 2 2 2 2 2 3 3 3 3	4

Here is one solution to assign students to four teams:

{1, 2}, {3, 4}, {5, 6}, {7, 8, 9}.

Notes

For instance, suppose there are five students ($N = 5$) with $X_1 = 1, X_2 = 2, X_3 = 5, X_4 = 2, \text{and} X_5 = 1$. In particular, student 3 wishes to be in a team with at most 5 members, whereas students 2 and 4 at most 2 students.

One (trivial) way to form teams is to let each student form a 1-person team – this yields five teams. Or, we can let students 2 and 4 form one team, and the other three students form 1-person teams – this results in four teams. In this example, there is no way to form 3 or fewer teams while meeting the said conditions.

Problem E. Eagle and Lambs

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 sec
Memory limit:	512 MB

Eagles eat lambs. The pasture on which lambs live can be described as a 1xN rectangle-shaped land, and it's divided into 1x1 cells. The cells are numbered as 1, 2, ... , N from left to right. In the i-th cell, there are A_i lambs living in it.

Every morning, an eagle hunts lambs. The eagle starts from the left-side of the first cell or the right-side of the right cell, and heads to the cell in which its target lamb lives. The eagle can only fly through the cells (directly on the pasture). If the target lamb is in the x-th cell, then it first flies to cell x, and it eats all of the lambs in it. The eagle can only eat lambs from one cell each day.

Lambs are scared of the eagle, so lambs begin to run away as soon as they see the eagle flying. Precisely, lambs run away when the eagle flies over the exact cell they live in. When the lambs run away from where the live, there will be no lambs left in that cell. For instance, if the eagle flies from the left-side of the first cell and arrive at cell x, then all of the lambs in cells $1, 2, \ldots, x-1$ would run away and there will not be any lambs left in those cells. Likewise, if the eagle flies from the right-side of the n-th cell, then all of the lambs living in cells $x+1, x+2, \ldots, N$ would run away.

In addition, one lamb from each cell goes missing (or gets eaten by other predators) every night.

The total number of lambs this eagle can hunt (and eat) depends on where it flies from (whether from the left-side or the right-side) as well as which cell it targets. Compute the maximum number of lambs this eagle can eat.

Input

The first line contains N, the number of cells ($1 \le N \le 1,000$).

The second line contains N integers $A_1, A_2, \ldots, A_N$, representing the number of lambs in each cell ($0 \le A_i \le 100,000$).

Output

The first line must contain the maximum number of lambs the eagle can eat.

Examples

standard input	standard output
5 1 10 4 10 1	21

On day 1, the eagle flies from the left and eats all lambs in cell 2; it would eat 10 lambs. After this, there will be no lambs left in cell 1 and 2. At night, each cell's one lamb goes missing (if any), so the cells 3 and 4 would have 3 and 9 lambs, respectively, and other cells will not contain any lambs.

On day 2, the eagle flies from the right, and eats the nine lambs in cell 4. The eagle has eaten 19 lambs at this point. At night one lamb from cell 3 goes missing, leaving only two lambs in it and no lambs in other cells.

On day 3, the eagle can eat the lambs in cell 3, totaling in 21 lambs.

There are many other ways to eat 21 lambs.

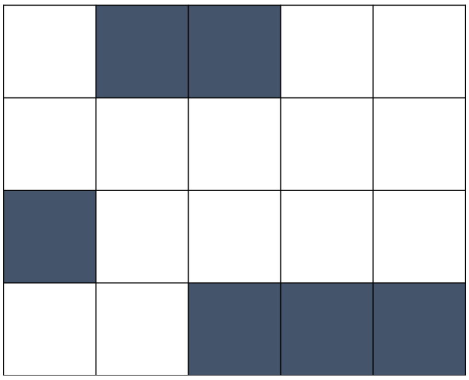
standard input	standard output
5 1 1 1 1 1	1
6 10 1 1 1 1 10	19
7 1 2 3 4 5 6 7	16
14 10 1 10 2 10 3 10 4 10 3 10 2 10 1	49
4 0 0 0 0	0

Problem F. Fun With Tiles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 sec
Memory limit:	512 MB

Albert got a lot of 1×2 tiles from a friend. This tile is shaped as two rectangular cells combined together, like the following:

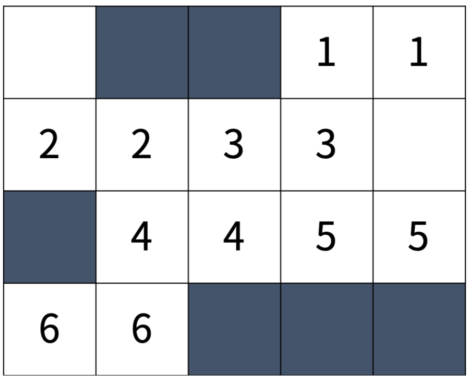
Coincidentally, Albert's vacation home balcony needs some repair – now he can use the tiles he got for free! The balcony's floor is a rectangle-shape which consists of $R \times C$ equal-size rectangle cells. The following example shows a sample with $R = 4$ and $C = 5$. Using one 1×2 tile, Albert can cover two horizontally adjacent rectangle cells completely. However he cannot rotate a tile and cover two vertically adjacent cells.



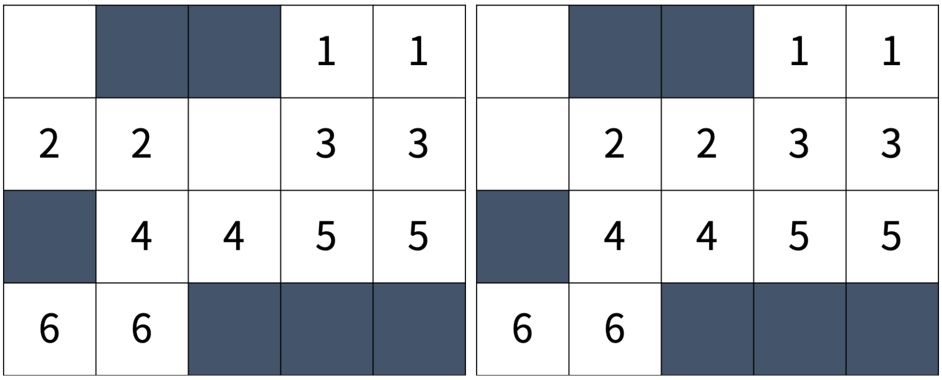
Because his vacation home hasn't been in use for a long time, a few cells have been damaged. In the example above, out of 20 cells, only six cells are undamaged (the grey ones) whereas the other fourteen cells are damaged and therefore need to be replaced. As Albert only has 1×2 tiles, he may not be able to cover all of the damaged cells, but he wishes to cover as many damaged cells as possible.

A single tile must cover two damaged cells completely, no tiles should overlap, no tile should go beyond the boundary of the balcony, and no tile should cover an undamaged cell.

In the sample example above, Albert can use six tiles to cover the most number of damaged cells. Although the tiles are identical (indistinguishable), for the sake of explanation, the six tiles are labeled in the following example.



There are two other ways to use six tiles to cover damaged cells.



Albert is curious to know in how many different ways he can cover damaged cells – while using as many tiles as possible. When Albert covers the floor in two different ways, we say that those two ways are "different ways" if there is at least one cell such that the cell is covered in one way but not in the other. If no such cells exist, then the two ways are the same because tiles are not distinguishable.

Input

The first line of input contains three integers R (the number of rows), C (the number of columns), and K (the number of undamaged cells). Assume that $1 \le R \le 1,000,000,000$, $2 \le C \le 1,000,000,000$ and $0 \le K \le 1,000$.

Each of the next K lines contains two integers (row number and column number) of an undamaged cell (rows are numbered 1 to R and columns from 1 to C). These K cells do not contain duplicates.

In addition, each test cases guarantees that at least one tile can be placed.

Output

You must output two integers. The first integer is the maximum number of tiles Albert can place, while meeting the conditions mentioned in the problem statement. The second number is the number of different ways in which Albert can place the maximum number of tiles. You can assume that the first number is at most 10^{18}. The second number can be very large, and therefore you must output this number modulo 1,000,000,007.

Examples

standard input	standard output
`2 2 0`	`2 1`

Each row requires one tile, and this is the unique way (to use maximum number of tiles).

standard input	standard output
`2 3 1` `2 2`	`1 2`

The first row can be filled with one tile (in two different ways), but the second row cannot be filled.

standard input	standard output
4 5 6 1 2 1 3 3 1 4 3 4 4 4 5	6 3
1000000000 1000000000 0	500000000000000000 1
1000000000 1000000000 2 324873289 23476755 99584832 4721222	499999999999999998 750063488

Problem G. Minimum Spanning Tree Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 sec
Memory limit:	512 MB

Consider a simple graph G that contains N vertices and M undirected edges. A simple graph requires that no self-loops exist and at most one edge exists between any pair of vertices. A spanning tree of this simple graph is a set of edges that satisfy the following conditions.

1. The set contains N-1 edges.
2. Given two arbitrary vertices, there must be a path between the two vertices only using the edges in this set.

Among spanning trees, a minimum spanning tree (MST) is a spanning tree whose sum of weights of edges is smallest.

We want to play the MST Game on a graph.

- In this game, we need to compute the cost of an MST as we remove edges one by one. The cost of an MST is the sum of weights of the edges in the MST. Each turn's score is equal to the cost of an MST found during that turn.
- The game is played for K turns, and in the first turn, we must compute the cost of MST of the given graph.
- At the end of each turn, we must remove the cheapest edge in the MST that we found during that turn.
- Once an edge is removed, it cannot be used again in subsequent turns.
- If an MST does not exist, the score for that turn is 0. For all subsequent turns, the score remains to be 0 as well. It is possible that no MST exists even in the first turn.

Given a simple graph (with undirected edges) and K, compute the score for each turn when this game is played.

Input

The first line contains the number of vertices N, the number of edges M, and the number of turns K ($2 \le N \le 1,000$, $1 \le M \le \min(10,000, N \times (N-1)/2)$, $1 < K \le 100$).

The following M lines describe M edges. Each line contains two vertices x and y. The same edge is given at most once. These M edges' costs are $1, 2, \ldots, M$ in the input order.

The vertices are labeled from 1 to N.

Output

You must output K numbers corresponding to the scores of K turns in a single line.

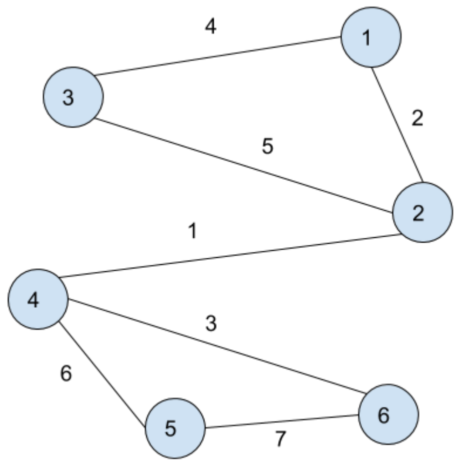
Examples

standard input	standard output
6 6 2 1 2 2 3 1 3 4 5 5 6 4 6	0 0

There is no MST in the given graph. Therefore all turns have score 0.

standard input	standard output
6 7 3 2 4 1 2 4 6 1 3 2 3 4 5 5 6	16 0 0
4 5 4 3 4 1 3 1 4 1 2 2 4	7 9 0 0
5 7 4 1 2 2 3 3 4 4 5 1 5 1 4 1 3	10 14 0 0
6 9 6 1 2 2 3 3 4 4 5 5 6 1 6 1 4 2 5 3 6	15 20 26 32 35 0

Notes



In this case, the MST found in the first turn contains five edges { (1, 3), (1, 2), (2, 4), (4, 6), (4, 5) } whose cost is 16. After we remove the edge (2, 4), we find that no MST exists in the second turn. Hence the score is 0 for turn 2 and after.

Problem H. Finding Tricky Numbers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 sec (No Language Bonus Time)
Memory limit:	512 MB

Albert recently learned about integers, digits, and subtraction. He's so fascinated by these new mathematical concepts, and he invented a game called the "Tricky Numbers".

In this game, he first picks an integer digit A (A is between 0 and 9, inclusive). He then defines "A-tricky numbers" as follows. A positive integer x is an A-tricky number if the following conditions are met:

1. If x only has one digit (i.e., x is between 1 and 9, inclusive), then it is an A-tricky number.
2. If x contains two or more digits, and the (absolute) difference between every pair of consecutive digits of x is greater than or equal to A, then x is an A-tricky number.

For instance, suppose Albert picks $A = 0$. Then every positive integer is a 0-tricky number.

When $A = 1$, the following list shows the first thirty 1-tricky numbers:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, ...

Albert wants to find the K-th smallest A-tricky number (given A and K), and to verify whether his findings are correct – he needs your help! Note that the number can be very large for certain values of K and A. Therefore, Albert says you can simply give him the K-th smallest A-tricky number modulo $(10^9 + 7)$.

Input

The first line will contain a positive integer, T ($1 \le T \le 100$), which is the number of test cases. Each of the following T lines will contain two positive integers (K and A), separated by a whitespace. K is between 1 and 10^{18} (inclusive) and A is between 0 and 9 (inclusive).

Output

For each test case, you should output the K-th smallest A-tricky number (modulo $(10^9 + 7)$). Each test case's output should be contained in a single line (see the sample output below).

Examples

standard input	standard output
8 5 0 5 1 30 1 12 2 20 5 21 5 30 8 1000 8	5 5 32 15 50 60 9190 80902435

- For the fourth test case ($K = 12$ and $A = 2$), note that 10, 11, and 12 are not 2-tricky numbers.
- For the last case, the actual answer is 808080908091 (before applying the modulo operation).