

Problem A. Who is The 19th ZUCCPC Champion

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

The 19-th ZUCCPC has started! Output anyone that you think he/she might be the champion.

Any string of length $n(1 \leq n \leq 100)$ consisting of only lowercase letters, uppercase letters, digits and spaces(the character ' ') will be considered correct.



Input

(no input)

Output

Output anyone might be the champion in a line.

Example

standard input	standard output
(no input)	Ranni the Witch

Problem B. Jiubei and Overwatch

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

It's high noon!

Jiubei loves playing Overwatch even if it is a long time since the last update. Now he is using a damage skill to kill enemies.

The skill will damage all enemies simultaneously and the damage is determined by the time Jiubei charges this skill. The skill has a charging period division number k , and two damage increase numbers x, y . If he charged for t seconds, the damage is calculated as follows:

- If $t \leq k$, the damage is $t \times x$.
- Otherwise, the damage is $k \times x + (t - k) \times y$

Jiubei has n enemies, and the i -th enemy's health is a_i .

Please tell Jiubei how long it takes to charge his skill to kill all of his enemies. **Note that the skill can only be used once.**

Additionally, an enemy is killed if and only if his health is **not** greater than the damage he/she take.

Input

The first line contains a single integer T ($1 \leq T \leq 100$) — the number of test cases.

The first line of each test case contains four integers n, k, x, y ($1 \leq n, k, x, y \leq 100$) — the number of enemies, the number of charging period division, the numbers of damage increase respectively.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — the health of enemies.

Output

For each test case, output a single integer in a line, indicating the **minimum** time Jiubei needs to charge his skill to kill all enemies.

Example

standard input	standard output
2	2
3 2 4 3	7
1 2 5	
3 2 4 3	
1 5 21	

Problem C. Ah, It's Yesterday Once More

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes

Boboge remembers The 2021 ICPC Asia Nanjing Regional Contest, Problem D. The problem gives you an array a and asks you how many times the expression “*Swap a_i and a_j* ” has been executed if you call $Sort(a)$ which is implemented by the following *Algorithm 1*.

Algorithm 1 The Sorting Algorithm

```

function SORT( $A$ )
  for  $i \leftarrow 1$  to  $n$  do                                 $\triangleright n$  is the number of elements in  $A$ 
    for  $j \leftarrow 1$  to  $n$  do
      if  $a_i < a_j$  then                                     $\triangleright a_i$  is the  $i$ -th element in  $A$ 
        Swap  $a_i$  and  $a_j$ 
      end if
    end for
  end for
end function

```

Algorithm 2 The Bubble Sorting Algorithm

```

function BUBBLESORT( $A$ )
  for  $i \leftarrow 1$  to  $n$  do                                 $\triangleright n$  is the number of elements in  $A$ 
    for  $j \leftarrow 1$  to  $n - 1$  do
      if  $a_j > a_{j+1}$  then                                   $\triangleright a_j$  is the  $j$ -th element in  $A$ 
        Swap  $a_j$  and  $a_{j+1}$ 
      end if
    end for
  end for
end function

```

As the algorithm looks quite like the normal bubble sort, Boboge wants you to construct a permutation of length n as the array a , so that the number of times the expression “*Swap a_i and a_j* ” has been executed equals to the number of times the expression “*Swap a_j and a_{j+1}* ” has been executed when we call $Sort(a)$ and $BubbleSort(a)$ respectively.

A permutation is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4, 5]$ is also not a permutation ($n = 4$ but there is 5 in the array).

Input

The first line contains a single integer T ($1 \leq T \leq 2 \times 10^5$) — the number of test cases.

Each test case consists of one line containing one integer n ($1 \leq n \leq 2 \times 10^5$) — the length of permutation that you need to construct.

It is guaranteed that $\sum n \leq 2 \times 10^5$ over all test cases.

Output

For each test case, print n integers in a line — the permutation you construct.

If there are multiple answers, print any. It can be proved that there is always a valid answer.

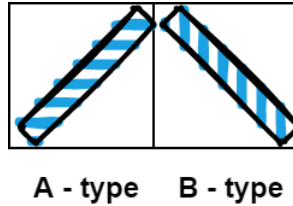
Example

standard input	standard output
2	1
1	2 1
2	

Problem D. Reflection

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

There are n mirrors on the grid numbered with integers from 1 to n . Mirrors have two types, 'A' and 'B', the two shapes are shown below. Both sides of the mirror can reflect rays. We define the bottom left corner as $(1, 1)$.



You need to answer q queries. The i -th query has two integers x_i, y_i and one letter $c_i \in \{'L', 'U', 'R', 'D'\}$, denotes a ray with (x_i, y_i) as the starting point and c_i as the direction. 'L', 'U', 'R', 'D' denotes *Left*, *Up*, *Right*, *Down* respectively. Changes of the coordinate corresponding to the direction are as follows:

- *Left*: $(x, y) \rightarrow (x - 1, y)$
- *Up*: $(x, y) \rightarrow (x, y + 1)$
- *Right*: $(x, y) \rightarrow (x + 1, y)$
- *Down*: $(x, y) \rightarrow (x, y - 1)$

For each query, find the **last** mirror reached by the ray.

You can refer to the note for a better understanding of the problem.

Input

The first line contains two integers n, q ($1 \leq n, q \leq 10^5$) — the number of mirrors and queries.

Then n lines follow, each line contains 2 integers X_i, Y_i and 1 character T_i — the coordinates and type of the i -th mirror. $1 \leq X_i, Y_i \leq 10^5$, $T_i \in \{'A', 'B'\}$.

Then q lines follow, each line contains 2 integers x_i, y_i and 1 character c_i — the starting point and direction of the ray in the i -th query. $1 \leq x_i, y_i \leq 10^5$, $c_i \in \{'L', 'U', 'R', 'D'\}$.

It is guaranteed that no two mirrors are in the same position and that for each query there are no mirrors at the starting point of the ray.

Output

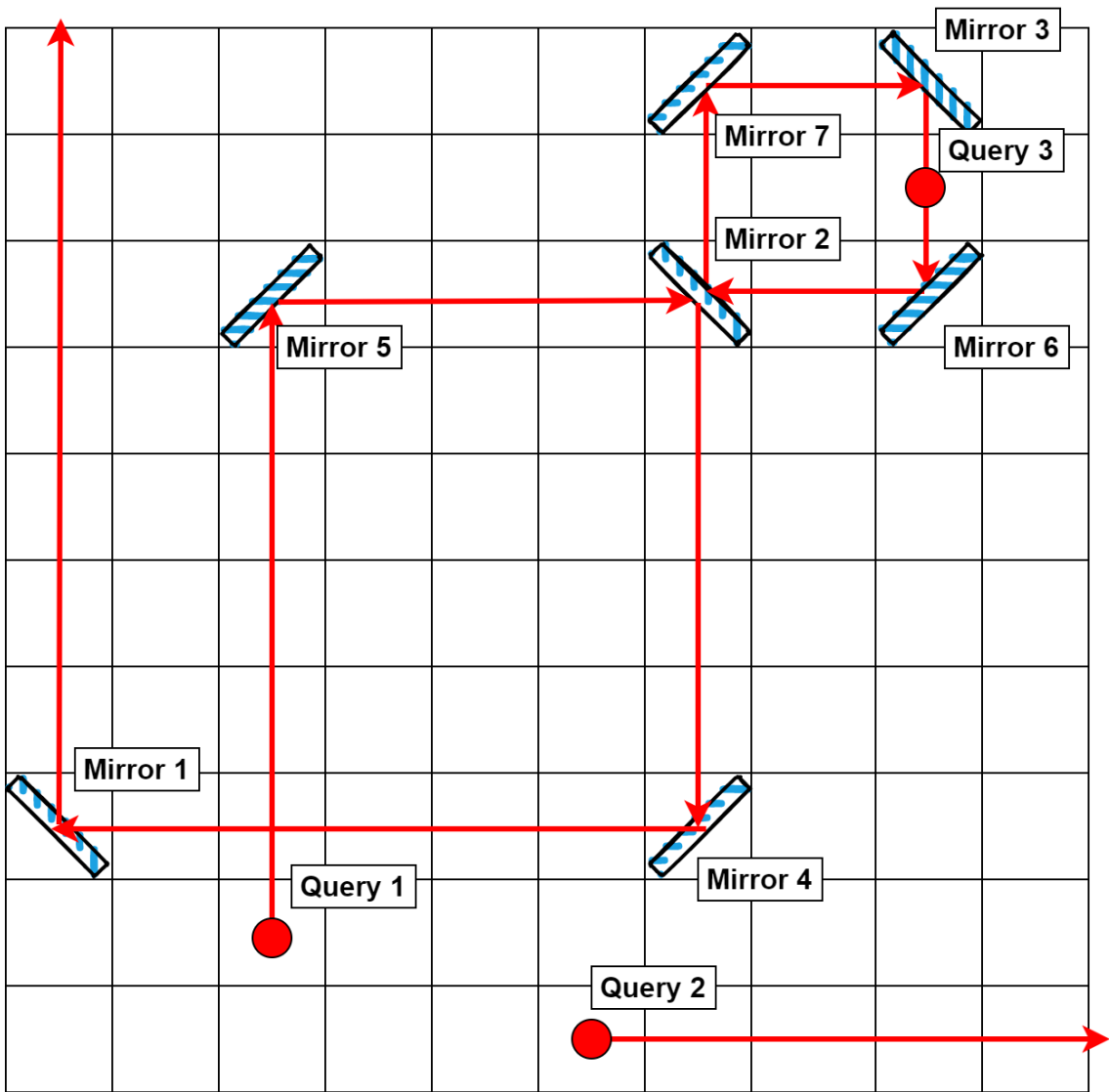
The output contains q lines, each line contains an integer — the **last** mirror reached by the ray. If the ray will reflect infinitely, print -1 . If the ray doesn't reach any mirror, print 0 .

Example

standard input	standard output
7 3	1
1 3 B	0
7 8 B	-1
9 10 B	
7 3 A	
3 8 A	
9 8 A	
7 10 A	
3 2 U	
6 1 R	
9 9 D	

Note

The grid of the sample is as follows.



Problem E. Disjoint Path On Tree

Input file: **standard input**

Output file: **standard output**

Time limit: 1 second

Memory limit: 256 megabytes

You are given a tree consisting of n vertices. Recall that a tree is an undirected connected acyclic graph.

Now, calculate the number of simple path pairs that each pair of simple paths does not pass through the same vertex. Note that $pair(Path_i, Path_j)$ and $pair(Path_j, Path_i)$ should only be calculated once. As the answer may be too big, output the answer module $10^9 + 7$.

The simple path is the path that visits each vertex at most once. The path only visits a single vertex is also a path.

Input

The first line contains a single integer $T (1 \leq T \leq 2 \times 10^5)$ — the number of test cases.

The first line of each test case contains a single integer $n (1 \leq n \leq 2 \times 10^5)$ — the number of vertices.

The next $n - 1$ lines of each test case describe the edges of the tree in form of $u_i, v_i (1 \leq u_i, v_i \leq n, u_i \neq v_i)$.

It is guaranteed that given graph is a tree.

It is guaranteed that $\sum n \leq 2 \times 10^5$ over all test cases.

Output

For each test case, output a single integer in a line — the answer module $10^9 + 7$.

Example

standard input	standard output
1 3 1 2 1 3	5

Note

5 pairs of paths in the sample are:

- $(1, 2)$
- $(1, 3)$
- $(2, 3)$
- $(1 - 2, 3)$
- $(1 - 3, 2)$

Problem F. Sum of Numerators

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

You are given 2 integers n and k , and there are n fractions as follows:

$$\frac{1}{2^k}, \frac{2}{2^k}, \frac{3}{2^k}, \dots, \frac{n}{2^k}$$

Please reduce each fraction to its **simplest form** and calculate the sum of the numerators.

A fraction $\frac{a}{b}$ is in simplest form if and only if $\gcd(a, b) = 1$, where $\gcd(a, b)$ denotes the greatest common divisor (GCD) of integers a and b . For example, the simplest form of $\frac{6}{4}$ is $\frac{3}{2}$, and the simplest form of $\frac{6}{2}$ is $\frac{3}{1}$.

Input

The first line contains an integer T ($1 \leq T \leq 10^5$) — the number of test cases.

Each test case consists of one line containing 2 integers n ($1 \leq n \leq 10^9$) and k ($0 \leq k \leq 10^9$).

Output

For each test case, output an integer in a line.

Example

standard input	standard output
2	15
5 0	12
5 1	

Note

In the second test case of the sample, the simplest form of the 5 fractions is as follows:

$$\frac{1}{2}, \frac{1}{1}, \frac{3}{2}, \frac{2}{1}, \frac{5}{2}$$

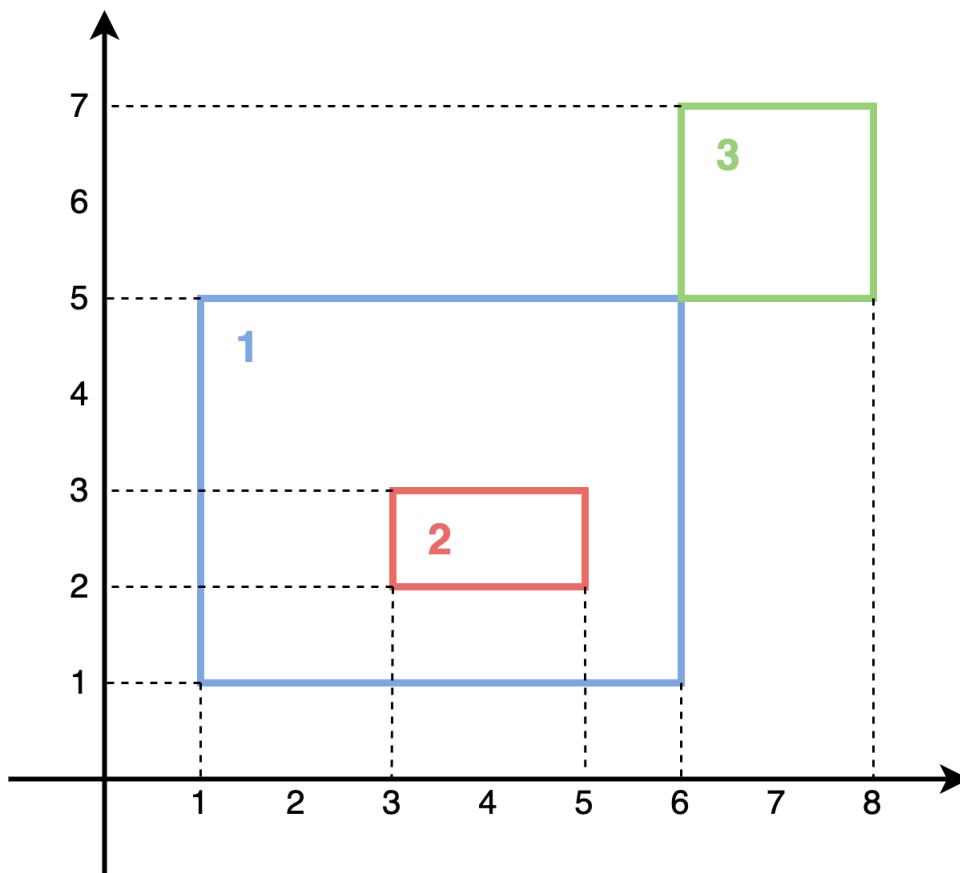
The sum of numerators is $1 + 1 + 3 + 2 + 5 = 12$.

Problem G. Guaba and Computational Geometry

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

There are n rectangles on a two-dimensional plane. The i -th rectangle has a weight w_i . You should find 2 non-overlapping rectangles i and j , and make $w_i + w_j$ as large as possible.

2 rectangles are non-overlapping if and only if no points are inside both rectangles at the same time. For example, rectangles 1 and 3 are overlapping, as are rectangles 1 and 2, but rectangles 2 and 3 are not overlapping.



Input

The first line contains an integer T ($1 \leq T \leq 3 \times 10^5$) — the number of test cases.

For each test case, the first line contains an integer n ($1 \leq n \leq 3 \times 10^5$) — the number of rectangles.

Then n lines follow, each line contains 4 integers a_i, b_i, c_i, d_i ($-10^9 \leq a_i, b_i, c_i, d_i \leq 10^9, a_i < c_i, b_i < d_i$) — the lower left corner (a_i, b_i) and upper right corner (c_i, d_i) .

The next line contains n integers w_i ($1 \leq w_i \leq 10^9$) — the weight of the i -th rectangle.

It is guaranteed that $\sum n \leq 3 \times 10^5$ over all test cases.

Output

For each test case, output an integer in one line — the maximum value of $w_i + w_j$. If all rectangles overlap each other, output -1 .

Example

standard input	standard output
3	3
3	-1
1 1 6 5	-1
3 2 5 3	
6 5 8 7	
10 1 2	
2	
1 1 6 5	
6 5 8 7	
1 2	
1	
-1 -1 1 1	
1000000000	

Problem H. Distance

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Here find a function $dist$ to calculate the distance from a point to an interval. For integer $l, r, x (l \leq r)$, we define $dist(l, r, x)$ as follows.

- If $x < l : dist(l, r, x) = l - x$
- If $l \leq x \leq r : dist(l, r, x) = 0$
- If $r < x : dist(l, r, x) = x - r$

You are given n pairs of integers, the i -th of which is (l_i, r_i) . It is guaranteed that $l_i \leq r_i$ for all i . For each $k = 1, 2, \dots, n$, solve the following problem.

- Choose an integer x arbitrarily to **minimize** $\sum_{i=1}^k dist(l_i, r_i, x)$, output the **minimum** value.

Input

The first line contains a single integer $n (1 \leq n \leq 2 \times 10^5)$ — the number of pairs.

Then n lines follow, each line contains two integers $l_i, r_i (-10^9 \leq l_i \leq r_i \leq 10^9)$ — the i -th pair you are given.

Output

Output n lines, the k -th line contains a single integer, which means the **minimum** value of $\sum_{i=1}^k dist(l_i, r_i, x)$. You can choose different x for different k .

Example

standard input	standard output
3	0
1 3	3
6 8	3
6 8	

Problem I. Array Division

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Zhongli gives you two arrays a and b of length n . He asks you to divide $[1, 2, \dots, n-1, n]$ into k continuous subarrays (a division for index), each index i belongs to exactly one subarray, and for these k continuous subarrays you divided, each meets the condition $\sum a_i \geq \sum b_i$.

Formally, you should find k pairs $[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k]$ satisfying following conditions:

- $l_1 = 1, r_k = n$
- $l_i \leq r_i$ for all $i \in [1, k]$
- $r_{i-1} + 1 = l_i$ for all $i \in [2, k]$
- $\sum_{j=l_i}^{r_i} a_j \geq \sum_{j=l_i}^{r_i} b_j$ for all $i \in [1, k]$

You should find the **maximum** k .

Input

The first line contains a single integer T ($1 \leq T \leq 5000$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 5000$) — the length of two arrays.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — the array a .

The third line of each test case contains n integers $b_1, b_2, \dots, b_n$ ($1 \leq b_i \leq 10^9$) — the array b .

It is guaranteed that $\sum n \leq 5000$ over all test cases.

Output

For each test case, output a single integer in a line — the **maximum** k , output -1 if there is no valid way to divide the two arrays.

Example

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

Problem J. Substring Inversion (Easy Version)

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 512 megabytes

This is the easy version of the problem. The only difference between the two versions is the constraint on n .

Given a string s of length n consisting of lowercase letters only, please calculate the number of 4-tuples (a, b, c, d) satisfying following conditions:

- $1 \leq a \leq b \leq n$
- $1 \leq c \leq d \leq n$
- $a < c$
- $s[a : b]$ is lexicographically greater than $s[c : d]$

Here, $s[i : j]$ represents a substring of s which start from the i -th character and end at j -th character(inclusive).

String x is lexicographically less than string y , if either x is a prefix of y (and $x \neq y$), or there exists such $i (1 \leq i \leq \min(|x|, |y|))$, that $x_i < y_i$, and for any $j (1 \leq j < i)$, $x_j = y_j$. Here $|a|$ denotes the length of the string a .

As the answer may be too big, please output the answer module $10^9 + 7$.

Input

The first line contains a single integer $T (1 \leq T \leq 10)$ — the number of test cases.

The first line of each test case contains a single integer $n (1 \leq n \leq 400)$ — the length of the string.

The second line of each test case contains a string s of length n consisting of lowercase letters only.

It is guaranteed that $\sum n \leq 400$ over all test cases.

Output

For each test case, output a single integer represents the answer module $10^9 + 7$.

Example

standard input	standard output
2	2
3	4
aab	
3	
aba	

Problem K. Substring Inversion (Hard Version)

Input file: standard input

Output file: standard output

Time limit: 3 seconds

Memory limit: 512 megabytes

This is the hard version of the problem. The only difference between the two versions is the constraint on n .

Given a string s of length n consisting of lowercase letters only, please calculate the number of 4-tuples (a, b, c, d) satisfying following conditions:

- $1 \leq a \leq b \leq n$
- $1 \leq c \leq d \leq n$
- $a < c$
- $s[a : b]$ is lexicographically greater than $s[c : d]$

Here, $s[i : j]$ represents a substring of s which start from the i -th character and end at j -th character(inclusive).

String x is lexicographically less than string y , if either x is a prefix of y (and $x \neq y$), or there exists such i ($1 \leq i \leq \min(|x|, |y|)$), that $x_i < y_i$, and for any j ($1 \leq j < i$), $x_j = y_j$. Here $|a|$ denotes the length of the string a .

As the answer may be too big, please output the answer module $10^9 + 7$.

Input

The first line contains a single integer T ($1 \leq T \leq 10$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \times 10^5$) — the length of the string.

The second line of each test case contains a string s of length n consisting of lowercase letters only.

It is guaranteed that $\sum n \leq 2 \times 10^5$ over all test cases.

Output

For each test case, output a single integer represents the answer module $10^9 + 7$.

Example

standard input	standard output
2	2
3	4
aab	
3	
aba	

Problem L. Monster Tower

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

To be The Elden Lord, The Tarnished will face many challenges. Here comes one.

There is a monster tower of height n , the strength of the monster on the i -th floor is a_i . The initial strength of The Tarnished is x .

For any moment, The Tarnished can choose some monster not stronger than him from the lowest k floors, and kill it. After The Tarnished kills the monster on the i -th floor ($1 \leq i \leq k$, $a_i \leq x$), his strength increases by a_i , and the i -th floor will disappear. So all floors higher than the i -th floor will fall, the original $(i + 1)$ -th floor become the i -th floor, $(i + 2)$ -th becomes $(i + 1)$ -th, and so on.

Now, the problem is what is the **minimum** x to kill all monsters in this tower.

Input

The first line contains a single integer T ($1 \leq T \leq 2 \times 10^5$) — the number of test cases.

The first line of each test case contains two integers n ($1 \leq n \leq 2 \times 10^5$) and k ($1 \leq k \leq n$) — the height of the tower and the highest floor that The Tarnished can reach respectively.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — the strength of each monster.

It is guaranteed that $\sum n \leq 2 \times 10^5$ over all test cases.

Output

For each test case, output a single integer — the **minimum** x to kill all monsters.

Example

standard input	standard output
2	4
3 2	9
1 10 5	
3 1	
1 10 5	

Note

In the first test case, The Tarnished kills the monster on the first floor, so his strength becomes 5, and the third floor becomes the second floor so The Tarnished can reach it and kill the monster. After that, his strength becomes 10 and can kill the last one.