

A. Progress Pie

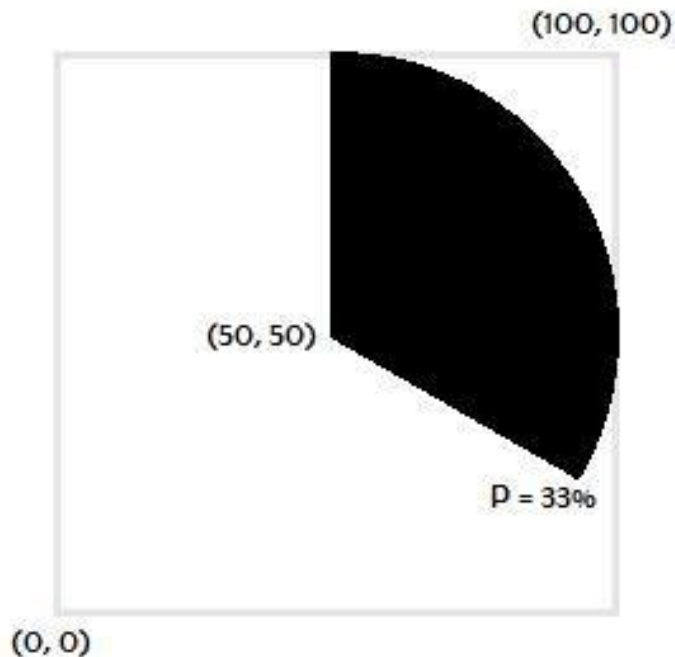
Input file: **standard input**
Output file: **standard output**
Time limit: 10 seconds
Memory limit: 1024 megabytes

Some progress bars fill you with anticipation. Some are finished before you know it and make you wonder why there was a progress bar at all.

Some progress bars progress at a pleasant, steady rate. Some are chaotic, lurching forward and then pausing for long periods. Some seem to slow down as they go, never quite reaching 100%.

Some progress bars are in fact not bars at all, but circles.

On your screen is a progress pie, a sort of progress bar that shows its progress as a sector of a circle. Envision your screen as a square on the plane with its bottom-left corner at $(0,0)$, and its upper-right corner at $(100,100)$. Every point on the screen is either white or black. Initially, the progress is 0%, and all points on the screen are white. When the progress percentage, P , is greater than 0%, a sector of angle $(P\% * 360)$ degrees is colored black, anchored by the line segment from the center of the square to the center of the top side, and proceeding clockwise.



While you wait for the progress pie to fill in, you find yourself thinking about whether certain points would be white or black at different amounts of progress.

Input

Input begins with an integer $T(1 \leq T \leq 1000)$, the number of points you're curious about. For each point, there is a line containing three space-separated integers, P , the amount of progress as a percentage, and X and $Y(0 \leq P, X, Y \leq 100)$, the coordinates of the point.

Output

For the i -th point, print a line containing "Case # i : " followed by the color of the point, either "black"

or 'white'.

Whenever a point (X, Y) is queried, it's guaranteed that all points within a distance of 10^{-6} of (X, Y) are the same color as (X, Y) .

Example

standard input	standard output
5	Case #1: white
0 55 55	Case #2: white
12 55 55	Case #3: black
13 55 55	Case #4: white
99 99 99	Case #5: black
87 20 40	

Note

In the first case all of the points are white, so the point at $(55, 55)$ is of course white.

In the second case, $(55, 55)$ is close to the filled-in sector of the circle, but it's still white.

In the third case, the filled-in sector of the circle now covers $(55, 55)$, coloring it black.

B. Lazy Loading

Input file: **standard input**
Output file: **standard output**
Time limit: 10 seconds
Memory limit: 1024 megabytes

Wilson works for a moving company. His primary duty is to load household items into a moving truck. Wilson has a bag that he uses to move these items. He puts a bunch of items in the bag, moves them to the truck, and then drops the items off.

Wilson has a bit of a reputation as a lazy worker. Julie is Wilson's supervisor, and she's keen to make sure that he doesn't slack off. She wants Wilson to carry at least 50 pounds of items in his bag every time he goes to the truck.

Luckily for Wilson, his bag is opaque. When he carries a bagful of items, Julie can tell how many items are in the bag (based on the height of the stack in the bag), and she can tell the weight of the top item. She can't, however, tell how much the other items in the bag weigh. She assumes that every item in the bag weighs at least as much as this top item, because surely Wilson, as lazy as he is, would at least not be so dense as to put heavier items on top of lighter ones. Alas, Julie is woefully ignorant of the extent of Wilson's lack of dedication to his duty, and this assumption is frequently incorrect.

Today there are N items to be moved, and Wilson, paid by the hour as he is, wants to maximize the number of trips he makes to move all of them to the truck. What is the maximum number of trips Wilson can make without getting berated by Julie?

Note that Julie is not aware of what items are to be moved today, and she is not keeping track of what Wilson has already moved when she examines each bag of items. She simply assumes that each bagful contains a total weight of at least $k \cdot w$ where k is the number of items in the bag, and w is the weight of the top item.

Input

Input begins with an integer T ($1 \leq T \leq 500$), the number of days Wilson "works" at his job. For each day, there is first a line containing the integer N ($1 \leq N \leq 100$). Then there are N lines, the i th of which contains a single integer, the weight of the i th item, W_i ($1 \leq W_i \leq 100$).

Output

For the i th day, print a line containing "Case # i : " followed by the maximum number of trips Wilson can take that day.

On every day, it is guaranteed that the total weight of all of the items is at least 50 pounds.

Example

standard input	standard output
5	Case #1: 2
4	Case #2: 1
30	Case #3: 2
30	Case #4: 3
1	Case #5: 8
1	
3	
20	
20	
20	
11	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
6	
9	
19	
29	
39	
49	
59	
10	
32	
56	
76	
8	
44	
60	
47	
85	
71	
91	

Note

In the first case, Wilson can make two trips by stacking a 30-pound item on top of a 1-pound item, making the bag appear to contain 60 pounds.

In the second case, Wilson needs to put all the items in the bag at once and can only make one trip.

In the third case, one possible solution is to put the items with odd weight in the bag for the first trip, and then the items with even weight in the bag for the second trip, making sure to put the heaviest item on top.

C. Crazy Dreamoon

time limit per test: 1 second

memory limit per test: 512 mebibytes

input: *standard input*

output: *standard output*

Dreamoon likes algorithm competitions very much. But when he feels crazy because he cannot figure out any solution for any problem in a competition, he often draws many meaningless straight line segments on his calculation paper.

Dreamoon's calculation paper is special: it can be imagined as the plane with Cartesian coordinate system with range $[0, 2000] \times [0, 2000]$ for the coordinates. The grid lines are all lines of the form $x = c$ or $y = c$ for every integer c between 0 and 2000, inclusive. So, the grid contains 2000×2000 squares.

Now, Dreamoon wonders how many grid squares are crossed by at least one of the lines he drew. Please help Dreamoon find the answer. Note that, to cross a square, a segment must contain an interior point of the square.

Input

The first line of input contains an integer N denoting the number of lines Dreamoon draw. The i -th line of following N lines contains four integers $x_{i1}, y_{i1}, x_{i2}, y_{i2}$, denoting that the i -th segment Dreamoon drew is a straight line segment between points (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}) .

- $1 \leq N \leq 2 \times 10^3$
- $0 \leq x_{i1}, y_{i1}, x_{i2}, y_{i2} \leq 2 \times 10^3$
- the lengths of all line segments in input are non-zero

Output

Output one integer on a single line: how many grid squares are crossed by at least one of the line segments which Dreamoon drew.

Examples

input
3 0 0 5 5 0 5 5 0 0 5 5 0
output
9
input
1 0 0 4 3
output
6
input
2 0 0 4 3 1 0 3 3
output
6

D. Dreamoon and NightMarket

time limit per test: 1 second

memory limit per test: 512 mebibytes

input: *standard input*

output: *standard output*

After moving to Taipei, Dreamoon always goes to dinner at Jinmei Night Market. There are N kinds of food sold at Jinmei Night Market. These foods are numbered from 1 to N . The price of one piece of i -th food is p_i .

Every night, Dreamoon will choose a non-empty set of foods and eat one piece of each food in this set. Dreamoon likes new things. So he won't choose the same set in two different nights. Besides this, because Dreamoon is a poor boy, each night, he will choose the cheapest food set that he didn't choose before.

Now, you are given a positive integer K . Can you tell Dreamoon which set of foods he will choose on K -th day? You only have to tell him how much he will spent on this day.

Input

The input consists of two lines. The first line contains an integer N . The second line consists of N integers p_1, p_2, \dots, p_N .

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq K \leq \min(10^6, 2^N - 1)$
- $1 \leq p_i \leq 10^8$

Output

Output one number indicating how much Dreamoon will spend on food on K -th day.

Examples

input
5 30 4 2 1 16 8
output
30
input
4 5 1 1 2 2
output
2

E. Zero Game

time limit per test: 1 second

memory limit per test: 512 mebibytes

input: *standard input*

output: *standard output*

You are given one string S consisting of only '0' and '1'. You are bored, so you start to play with the string. In each operation, you can move any character of this string to some other position in the string. For example, suppose $S = '0010'$. Then you can move the first zero to the tail, and S will become '0100'.

Additionally, you have Q numbers K_1, K_2, \dots, K_Q . For each i , you wonder what can be the maximum number of consecutive zeroes in the string if you start with S and use at most K_i operations. In order to satisfy your curiosity, please write a program which will find the answers for you.

Input

The first line of input contains one string S . The second line of input contains one integer Q . Each of the following Q lines contains one integer K_i indicating the maximum number of operations in i -th query.

- $2 \leq N \leq 10^6$
- the length of S is exactly N characters
- S consists of only '0' and '1'
- $1 \leq Q \leq 10^5$
- $N \times Q \leq 2 \times 10^7$
- $1 \leq K_i \leq 10^6$

Output

For each query, output one line containing one number: the answer for this query.

Example

input
0000110000111110
5
1
2
3
4
5
output
5
8
9
9
9

F. Hacker Cups and Balls

time limit per test: 4 seconds
 memory limit per test: 512 mebibytes
 input: *standard input*
 output: *standard output*

The dark side of Dreamoon, Admenoor, wants to play an interesting game with you. Did you hear about the game "cups and balls"? Here is the hacker version of it!

There are n cups and n balls, both are numbered $1, 2, \dots, n$. At each moment of time, there is exactly one ball in each cup. Initially, a_i -th ball is placed in the i -th cup. Admenoor will perform m magic operations on these balls and cups. The i -th operation will sort all the balls in the cups numbered between l_i and r_i , inclusive. The sorting could be performed in either ascending order or descending order. After these m operations, you need to answer which ball is placed in the center cup. We guarantee that n will be an odd integer, so the center cup means the $\frac{n+1}{2}$ -th cup.

For example, consider $n = 5$, $m = 2$ and $a = [5, 1, 4, 2, 3]$. If the first operation is to sort the balls in the cups numbered between 1 and 4 in ascending order, then a would become $[1, 2, 4, 5, 3]$. If the second operation is to sort the balls in the cups numbered between 2 and 5 in descending order, then a would become $[1, 5, 4, 3, 2]$. In this example, the number of the ball in the center cup after all operations is 4.

Input

The first line of input contains two integers n and m . The following line contains n integers a_1, a_2, \dots, a_n . Each of the following m lines contains two integers l_i and r_i . If $l_i < r_i$, Admenoor will sort that balls in ascending order in this operation; otherwise, the balls will be sorted in descending order in this operation.

- $1 \leq n \leq 99\,999$
- n is an odd integer
- $0 \leq m \leq 10^5$
- $1 \leq a_i \leq n$
- $\langle a_i \rangle$ is a permutation of $1, 2, \dots, n$
- $1 \leq l_i, r_i \leq n$

Output

Output a single line with the number of the ball in the center cup after all operations.

Examples

input
3 2 1 3 2 1 3 3 1
output
2
input
5 2 5 1 4 2 3 1 4 5 2
output
4