A Median Person

1.0s, 256MB

Three students each took an exam and received a score. Every score is a different integer, so exactly one student scored higher than exactly one other and lower than the remaining one — that student's score is the **median**: it is neither the highest nor the lowest of the three.

Given the three students' names and their scores, determine the name of the student who scored the median.

INPUT

The first line contains three names, separated by single spaces. Each name is a non-empty string of lowercase English letters.

The second line contains three integers, separated by single spaces: the scores of the three students in the same order as the names. Each score is between 0 and 100, and the three scores are pairwise distinct.

OUTPUT

Print a single line containing the name of the student whose score is the median of the three.

EXAMPLE

INPUT	OUTPUT
`ann bob cid` `5 2 8`	`ann`

INPUT	OUTPUT
`x y z` `8 5 2`	`y`

INPUT	OUTPUT
`p q r` `2 8 5`	`r`

NOTE

In the first sample the scores are ann = 5, bob = 2, cid = 8; the median score 5 belongs to ann.

In the second sample x scored the highest and z the lowest, so y is the median.

B Holy Triangle Talisman

1.0s, 256MB

Jack bought three holy bamboos from Princess Kaguya at different times. Each bamboo initially had a length of A units. Since then, the bamboos have experienced 0, 1, and 2 moonrises, respectively. Whenever a moonrise occurs, a bamboo absorbs magical power and grows by exactly B units.

To fend off the invasion of the mage Jeffrey using a Holy Circle Talisman, Jack must arrange the three bamboos so that they form a triangle. To do so, he needs to connect their tips together. Determine whether the current lengths of the three bamboos can form a triangle.

Recall that three lengths can form a triangle if and only if the sum of any two lengths is **strictly greater** than the remaining length.

INPUT

The only line contains two integers A and B ($1 \le A, B \le 500$).

OUTPUT

Print Yes if the three bamboos can form a triangle, or No otherwise.

EXAMPLE

INPUT	OUTPUT
`3 1`	`Yes`

INPUT	OUTPUT
`1 2`	`No`

INPUT	OUTPUT
`2 2`	`No`

NOTE

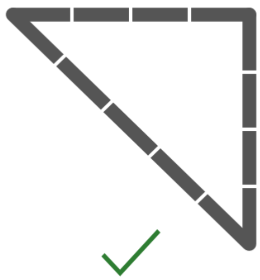


Fig.1

Fig.2A

Fig.2B

In example case 1, Jack's bamboo lengths are 3, 4, and 5, respectively. He can form a triangle as Fig.1 illustrates.

In example case 2, Jack's bamboo lengths are 1, 3, and 5, respectively. As Fig.2A suggests, Jack cannot form a triangle by connecting their tips. Creating a triangle as in Fig.2B is also invalid since the tips of the bamboo with lengths 3 and 5 do not connect.

C Movie Seat

1.0s, 256MB

You have been invited to the premiere of a new film directed by your childhood friend. They sent you an exclusive opening-night ticket to the screening at the royal theater. The film itself is an absurdly chaotic epic:

A magical soldier and a wealthy courtier wage a chaotic, decades-long romantic tug-of-war over a childhood friend. What begins as a petty rivalry rapidly escalates into a messy saga of corpse-carving sorcery, ghost-powered murders, family feuds, pitch-black court intrigue, and full-scale war. After a long series of abductions marriages and escapes, the love triangle comes to a sudden halt when the heroine got executed. Yet the chaos rages on: a second generation of sons takes over the feud, sparking witch-doctor love spells, fake rebel invasions, and a climax showdown with a vengeful, shape-shifting crocodile.

(Un)fortunately, a highway accident delayed your arrival at the theater. By the time you reach the main auditorium, everyone has already taken their seats.

The front row of the theater consists of seats arranged in a straight line. Each seat is represented by a character in a string:

- x indicates that the seat is already occupied by another guest.
- o indicates that the seat is still empty.

Since you want to enjoy the movie without disturbing anyone around you, you do not want to sit next to another guest. In other words, you may only choose an empty seat whose adjacent seats (if they exist) are also empty.

Determine how many seats you can choose to sit on.

INPUT

The input consists of a single string S ($1 \le |S| \le 5000$), representing the arrangement of seats in the theater's front row.

OUTPUT

Print a single integer: the number of seats where you can sit.

EXAMPLE

INPUT	OUTPUT
`ooxooooooxo`	3

INPUT	OUTPUT
`ooooo`	5

INPUT	OUTPUT
xoxx	0

NOTE

- For the first testcase, seats numbered 1, 5, and 6 are valid choices. Therefore, the answer is 3.
- For the second testcase, the entire row is empty, so every seat is a valid choice. Therefore, the answer is 5.
- For the third testcase, the only empty seat is next to an occupied seat. Therefore, there is no valid choice, and the answer is 0.

D Leftover

1.0s, 256MB

Captain Tsubasa and his wingless crew have finally uncovered the legendary buried treasure of Eno Island! Deep inside a secret cave, they discovered several heavy treasure chests, each containing a certain number of gold coins.

The pirates were thrilled to haul the loot back to their ship, but they quickly ran into a problem: they are short by exactly one crew member! Each pirate can carry at most one chest, and because they are short-staffed, exactly one treasure chest must be left behind on the island.

Furthermore, the gold coins cannot be transferred out of their original chests or split between crew members.

As the quartermaster, your job is to calculate the maximum total number of gold coins the crew can carry back to the ship.

INPUT

The first line contains an integer N ($2 \le N \le 100$), the total number of treasure chests.

The second line contains N integers A1, A2, ..., AN, where Ai ($1 \le Ai \le 1000$) is the number of gold coins in the i-th chest.

OUTPUT

Print a single integer: the maximum total number of gold coins that the N - 1 crew members can carry back to the ship.

EXAMPLE

INPUT	OUTPUT
5 3 6 7 4 6	23

NOTE

Since there are only 5 - 1 = 4 crew members to carry treasure, the possible total number of gold coins carried are:

- Leave the 3 gold coin chest: 6 + 7 + 4 + 6 = 23
- Leave the first 6 gold coin chest: 3 + 7 + 4 + 6 = 20
- Leave the 7 gold coin chest: 3 + 6 + 4 + 6 = 19
- Leave the 4 gold coin chest: 3 + 6 + 7 + 6 = 22
- Leave the second 6 gold coin chest: 3 + 6 + 7 + 4 = 20

Therefore, the maximum number of gold coins that the crew can carry back to the ship is 23 coins.

E No lemon, no melon

1.0s, 256MB

Hannah is a math professor who is obsessed with palindromes. For her birthday, you and your friend want to prepare a special gift for her.

Your friend has prepared A lemons. You want to buy B melons, but Hannah will only be impressed if the total number of fruits is a palindrome.

A palindrome is a number that reads the same forwards and backwards. For example, 767, 1331, and 999 are palindromes, while 69, 123, and 110 are not.

Your task is to find the number of melons you should buy so that the total number of fruits becomes a palindrome.

INPUT

The input contains a single integer A ($1 \le A \le 10^{12}$), the number of lemons prepared by your friend.

OUTPUT

Print a single integer B ($1 \le B \le 10^{12}$), the number of melons you should buy.

If there are multiple valid answers, print any of them.

It is guaranteed that a valid B always exists within the given constraints.

EXAMPLE

INPUT	OUTPUT
67	10

INPUT	OUTPUT
998244353	1755646

INPUT	OUTPUT
12345678910	55331088766

NOTE

For the first testcase, one possible answer is 10, since 67 + 10 = 77 is a palindrome.

For the second testcase, one possible answer is 1755646, since 998244353 + 1755646 = 999999999 is a palindrome.

For the third testcase, one possible answer is 55331088766, since 12345678910 + 55331088766 = 67676767676 is a palindrome.

F Upscaling

1.0s, 256MB

While looking through the University's old files, you found images of old University events from long ago. You want to display it on your monitor, but the images are low resolution, and so you decided to upscale them.

The images has resolution $N \times M$ (meaning, they are a grid of N by M pixels). You chose an integer K to upscale by. Each pixel in the original image then becomes $K \times K$ in the upscaled image, and the entire image becomes $N \cdot K \times M \cdot K$ in size.

In order to limit the data storage on the server (there really are a lot of images, and sometimes you upscaled the images by thousands), you decide that you'll crop the image, and also compute how many distinct pixels from the original image are in the cropped image, so that you know how much data you'll use. You crop the image so that it contains row r1 to r2 and column c1 to c2 of the upscaled image.

Your job now is to compute the number of distinct pixels from the original image in the cropped image. Two pixels are distinct if and only if they originated from different pixels in the original image. You must process T independent images.

INPUT

Each test contains multiple test cases. The first line contains a single integer T ($1 \le T \le 10^4$) — the number of images (testcases).

Each testcase contains 7 integers, N, M, K, r1, c1, r2, c2, which are the size of the images (row size, column size), the upscaling factor, the left row, top column, right row and bottom column of the cropping, respectively.

The integers satisfy $1 \le N, M, K \le 30000$, $1 \le r1 \le r2 \le N \cdot K$, $1 \le c1 \le c2 \le M \cdot K$

OUTPUT

For each test case, print a single integer — the number of distinct pixels from the original image in the cropped image.

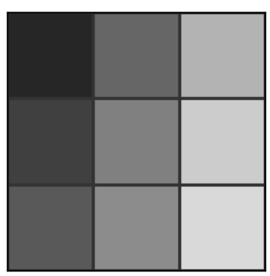
ICPC 2026 Thailand (Southern Region)

EXAMPLE

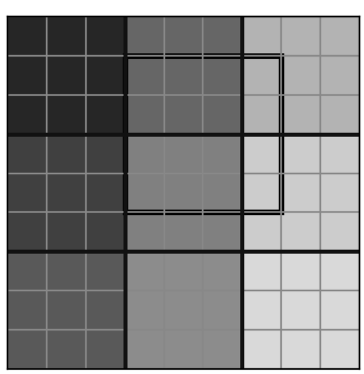
INPUT	OUTPUT
16	4
3 3 3 2 4 5 7	9
3 3 3 1 1 9 9	1
3 3 3 1 1 1 1	1
3 3 3 1 1 3 3	4
3 3 3 3 3 4 4	25
5 5 1 1 1 5 5	9
5 5 1 2 3 4 5	1
1 1 30000 1 1 30000 30000	1
1 1 30000 15000 1 20000 30000	1
100 100 7 50 50 50 50	4
10 10 5 6 6 15 15	4
10 10 5 5 5 6 6	900000000
30000 30000 3000 1 1 90000000 90000000	30000
30000 1 1 1 1 30000 1	25
5 5 7 3 4 30 29	21
3 7 4 1 1 12 28	

NOTE

Consider the first five testcases, which have N = M = K = 3. This means the original image is 3 × 3, like this. (Colors for illustrative purposes)

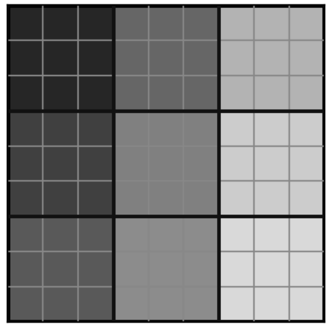


Then, it is upscaled by K = 3, to look like this

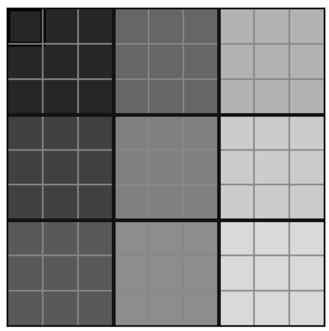


For every testcase, the black lines indicate the 'cropped image' region.

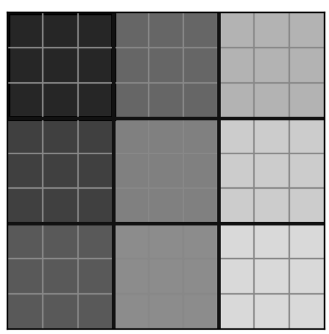
For the first testcases, it is cropped from row 2 to 5 and column 4 to 7, so it contains 4 original pixels.



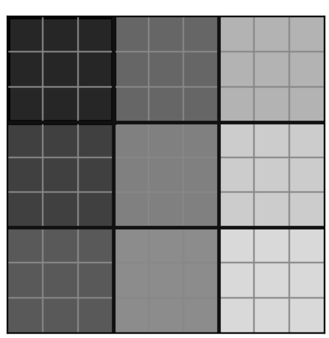
For the second testcases, it is cropped from row 1 to 9 and column 1 to 9 (the whole image), so it contains 9 original pixels.



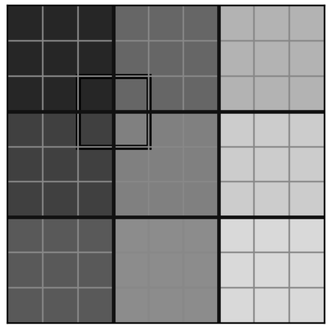
For the third testcases, it is cropped from row 1 to 1 and column 1 to 1 (a single small pixel), so it contains 1 original pixel.



For the fourth testcases, it is cropped from row 1 to 3 and column 1 to 3 (the whole pixel from the original image), so it contains 1 original pixel.



For the fifth testcases, it is cropped from row 3 to 4 and column 3 to 4 (each corner of 4 original pixels), so it contains 4 original pixels.



G EV Charging

2.0s, 256MB

An EV charging station can install any number of charging heads, but each head that is opened costs money, so the station wants to open as few as possible while still honoring every reservation made today.

There are N reservation requests, numbered from 1. The i-th request asks to charge for exactly one hour starting at time ti. Because of a hardware limitation, a single charging head cannot be used twice in a row if the two requested start times are D hours apart or less. Precisely: if a head serves some request starting at time a, and the very next request it serves (in time order) starts at time b > a, then it must hold that b - a > D. A head may serve any number of requests over the course of the day, and every request must be assigned to exactly one head.

Requests are not necessarily given in sorted order of start time, and two or more requests may ask for the exact same start time — such requests can never share a head, no matter how large D is (their gap is 0, which is always ≤ D).

Determine the minimum number of charging heads the station must open so that every request can be assigned to some head without violating the cooldown rule.

INPUT

The first line contains two integers N and D — the number of requests and the cooldown length, in hours.

The second line contains N integers t1, t2, ..., tN — the requested start times.

- $1 \le N \le 2 \cdot 10^5$
- $0 \le D \le 10^9$
- $0 \le ti \le 10^9$
- The ti are not necessarily distinct, and not necessarily given in sorted order.

OUTPUT

Output a single integer: the minimum number of charging heads required.

EXAMPLE

INPUT	OUTPUT
6 2 3 1 2 8 7 3	4

INPUT	OUTPUT
4 5 100 1 50 3	2

NOTE

In Sample 1, sorted start times are 1, 2, 3, 3, 7, 8, with D = 2 (so two uses of the same head must be more than 2 hours apart, i.e. at least 3 hours). One optimal plan: head 1 serves the requests starting at 1 and 7 (gap 6 > 2); head 2 serves 2 and 8 (gap 6 > 2). The two requests that both start at 3 cannot share a head with each other, and neither fits onto head 1 or head 2 in time, so each needs its own head. That is 4 heads in total, and no assignment does better.

In Sample 2, sorted start times are 1, 3, 50, 100, with D = 5 (gap must be at least 6). Head 1 serves 1 then 50 (gap 49 > 5); head 2 serves 3 then 100 (gap 97 > 5). Two heads suffice.