The International Collegiate Programming Contest
Sponsored by ICPC Foundation

The 2024 Baath Collegiate Programming Contest

(Contest Problems)

Syria
June 2024

Problem A. Boring Class

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Bronze`

Once upon a time in a boring class of electronic systems, *Kifah* and *Ali* were sitting there talking about the problem set of Al-Baath contest. Kindhearted *Ali* came up with a cute probability problem and told *Kifah* about it. Since *Kifah* doesn't like cute problems, he couldn't prevent himself from making some changes to make it a little bit harder. And here is the problem.

In this problem, you will be given an array $a_1, a_2, ..., a_n$. Imagine an array $b_1, b_2, ..., b_n$ where $b_i \in [1, a_i]$ with a uniform distribution. A subarray of b is one of its contiguous subsequences (i.e. an array $[b_l, b_{l+1}, ..., b_r]$ for some integers l and r such that $1 \le l \le r \le n$).

We define the function $f(l, r, x)$ of the subarray $[b_l, b_{l+1}, ..., b_r]$ as the probability of $max([b_l, b_{l+1}, ..., b_r]) \le x$.

You will be asked two types of queries:

- Type 1: replace a_p with v for the given integers p and v.
- Type 2: find $f(l, r, x)$ for the given integers l, r, and x.

It can be shown that this answer can always be expressed as a fraction P/Q where P, Q are coprime integers and $Q \neq 0 \bmod 10^9 + 7$. Output the value of $(P \cdot Q^{-1})$ modulo $10^9 + 7$.

Input

The first line contains a single integer Tc ($1 \le Tc \le 10^5$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the size of the array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$).

The third line of each test case contains a single integer q ($1 \le q \le 10^5$) — the number of queries.

Each of the following q lines has one of two types:

- Type 1: $1\ p\ v$ where ($1 \le p \le n$, $1 \le v \le 10^5$) — the position of the replaced integer and the new value, respectively.
- Type 2: $2\ l\ r\ x$ where ($1 \le l \le r \le n$, $1 \le x \le 10^5$) — the required parameters to define $f(l, r, x)$.

It is guaranteed that the sum of n and q over all test cases doesn't exceed 10^5.

Output

For each query of the type 2, print $f(l, r, x)$.

Examples

standard input	standard output
1 3 1 2 3 2 1 2 6 2 1 3 3	500000004
1 5 32 45 12 43 36 5 2 1 5 17 2 2 4 13 1 3 1000 2 1 4 40 2 2 5 55	242668409 929198973 340051682 135000001

Problem B. Best Substring

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Light Green`

You are given three integers x, y, z and a string S consisting of n lowercase English letters.

For four integers l_1, r_1, l_2, r_2 where $(1 < l_1 \leq r_1 < l_2 \leq r_2 < n)$, let's define the function F as follows:

$$F(l_1, r_1, l_2, r_2) = (r_1 - l_1) \times x + (r_2 - l_2) \times y + (l_2 - r_1) \times z + x + y - z$$

You need to find four integers l_1, r_1, l_2, r_2 where $(1 < l_1 \leq r_1 < l_2 \leq r_2 < n)$ that maximize the value $F(l_1, r_1, l_2, r_2)$ and satisfy the following conditions:

- The substring $S[l_1...r_1]$ is equal to a prefix of the string S.
- The substring $S[l_2...r_2]$ is equal to a suffix of the string S.

Note that $S[l...r]$ is a substring of the string S that starts from position l and ends at position r, more formally $S[l...r] = S_l S_{l+1} ... S_{r-1} S_r$, and if $l > r$ then $S[l..r]$ is an empty string.

If there are no integers l_1, r_1, l_2, r_2 satisfying the mentioned conditions, print 0.

Input

The first line contains four integers n, x, y, z $(4 \leq n \leq 10^6, 1 \leq x, y, z \leq 10^9)$ — where n is the number of letters in the string S.

The second line contains the string S, consisting of n lowercase English letters.

Output

Output a single integer — the maximum possible value of the function F, and if there are no integers l_1, r_1, l_2, r_2 satisfying the mentioned conditions, print 0.

Examples

standard input	standard output
`14 5 10 1` `abreabqqytpsyt`	`32`
`5 2 3 5` `aaaaa`	`10`

Note

In the first test, the four integers are $l_1 = 5, r_1 = 6, l_2 = 9, r_2 = 10$.

$S[5...6] = ab$ which equals to a prefix of S.

$S[9...10] = yt$ which equals to a suffix of S.

$(6 - 5) * 5 + (10 - 9) * 10 + (9 - 6) * 1 + 10 + 5 - 1 = 32$.

Problem C. Salameh Leveling

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Purple`

You are given a grid of $(n * m)$ points , each point contains a hidden box. When you collect it, you either gain some experience points ***"XPs"*** , or you lose some for being a horrible greedy player.

The grid has a guardian who won't let you take more than k boxes. Moreover, each n points forming a column has its own guardian. This guardian won't let you open more than i boxes in the column numbered i (in column number x, you can collect at most x boxes).

You want to level-up, so you want to gather as much ***XPs*** as possible.

* Columns are indexed 1 based (i : 1 2 3 ... m)

Input

The first line contains three integers n, m, and k $(1 \le n * m \le 10^6), (0 \le k \le n * m)$, the dimensions of the grid and the maximum number of boxes you can take.

Follow n lines, each containing m separated integers: $a_{i1}, a_{i2}, ..., a_{im}$ $(-10^9 \le a_{ij} \le 10^9)$, the value of each box in the grid.

Output

Print a single integer, the maximum amount of ***XPs*** you can get.

Examples

standard input	standard output
`1 1 1` `1000`	`1000`

`5 2 10`
`8 6666`
`5 1000`
`6 8457`
`9699 88889`
`125 0`
	`107045`

`3 3 6`
`2 -1 5`
`-2 3 3`
`4 5 6`
	`26`

Problem D. Math Game

Input file: standard input
Output file: standard output
Balloon Color: Orange

Aghiad and Aram love math so much. They consider playing video games, sports, and any other activity rather than math a waste of time. But sometimes they get bored from their math problems and want to do something fun without wasting their time. To kill boredom, they came up with a math game that will let them have some fun and waste no time. They write down an array consisting of n integers. They play in turns, in each turn:

1. First, Aghiad can choose an element at position i $(1 \le i < n)$.
2. Then Aram can choose an element at position j $(i < j \le n)$.

Now they check if they both scored a point (they are not playing against each other, they are playing together). In this turn, they score a point if and only if the pair they have chosen fulfills this condition:

$$\frac{a_i^3+i^3}{a_j^3+j^3} = \frac{a_i+i}{a_j+j} \text{ and } 1 \le i < j \le n$$

If they score a point, they add it to the total and write down on a paper the pair (i, j) so they can remember it. They cannot choose any pair that was chosen earlier. The game finishes when there are no pairs to choose from. In one day, they wrote an array a of length n, and they were too busy to play this game, but they were too curious about the total score of this array, so they asked you to tell them the final score.

Input

The first line of the input contains a single integer t $(1 \le T \le 10^4)$ — the number of test cases in the test.

The first line of each case contains a single integer n $(2 \le n \le 2 \cdot 10^5)$ — the number of elements in the array.

The second line of each case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_j \le 10^9)$ — the elements of the array.

It is guaranteed that the sum of the values of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the score that Aghiad and Aram would achieve if they play the game.

Example

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

Note

In the first testcase:

$$\frac{a_1^3+1^3}{a_2^3+2^3} = \frac{2^3+1^3}{1^3+2^3} = 1 = \frac{2+1}{1+2} = \frac{a_1+1}{a_2+2}$$

In the second testcase, all the pairs satisfy the condition of getting a point.

Problem E. Hard Test

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Rose`

Kifah heard that some contestants in Al-Baath University doesn't know how to read an integer in c++. So he prepared a hard test to find out.

you will read an integer n, so we make sure you are a good programer. Then, you can ask the jury the verdict you want. You can print "AC" (without quotes) if you want "Accepted" verdict, or you can print any thing else.

Input

The only line has one integer n ($1 \leq n \leq 1000$) some nonsense integer.

Output

Print "AC" (without quotes) if you want "Accepted" verdict. Otherwise, print any thing.

Hint: You are here to get Accepted.

Example

standard input	standard output
5	AC

Problem F. Good Friend

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Silver`

Abdullah and his friend $Kifah$ live in a real country called BP, which has n cities that can be described as a weighted tree rooted at node 1.

$kifah$ wants to buy some presents for his girlfriend but unfortunately he doesn't have any money, so he will borrow some money from his friend *Abdullah*.

Abdullah is a good friend, so he will help $Kifah$.

Abdullah works as a TAXI driver. For a ride from city a to city b (note that city b must be in the subtree of the a, more formally the city a must lie on the simple path between city 1 and city b), *Abdullah* earns an amount of money equal to the sum of weights in the simple path between a and b , and he spends hours equal to the number of edges in the simple path between a and b.

$Kifah$ loves his girlfriend so much and wants to buy her q presents.

Abdullah will give you the construction of the country and then for each present he will tell you the city $city_i$ that he is currently located in and the price p_i of the present that $Kifah$ wants to buy. Can you tell him the minimum possible number of hours required to make money more than or equal to p_i. If it's impossible to earn that amount of money just print -1.

Input

The first line contains an integer $n (1 \leq n \leq 10^5)$, the number of the cities in BP.

The next $n - 1$ lines describe the roads. The i_{th} line contains three integers u_i, v_i and w_i $(1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^9)$, meaning that there is an edge between the nodes u_i and v_i with weight w_i.

The next line contains a single integer q $(1 \leq q \leq 10^5)$, the number of presents that $Kifah$ wants to buy for his girlfriend.

For the following q lines, the i_{th} line contains two integers $city_i, p_i$ $(1 \leq city_i \leq n, 1 \leq p_i \leq 10^{18})$ meaning that *Abdullah* will start from $city_i$ and wants to earn an amount of money more than or equal to p_i.

Output

For each present, print the minimum possible number of hours required to make money more than or equal to p_i if *Abdullah* started the ride from $city_i$, or if it's impossible to earn that amount of money, just print -1.

Example

standard input	standard output
5 1 2 3 2 3 4 2 4 2 3 5 6 3 1 6 2 11 5 2	2 -1 -1

Note

To buy the first present, *Abdullah* can go from city 1 to city 3 spending 2 hours, and earning an amount of money equal to 7 which is greater than 6.

For the second present, the maximum amount of money *Abdullah* can earn is 10 by having a ride to the city 5, 10 is lower than 11, so the answer is -1.

For the third present, *Abdullah* can't go to any city (because there are no cities in the subtree of the city 5), so the answer is -1.

Problem G. Da7doo7

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Yellow`

Da7doo7 believed he had mastered competitive programming contests, so he decided to turn his attention to playing basketball.

Da7doo7 intends to shoot n balls into the basket. The probability of him scoring the first shot is $\frac{p_0}{q_0}$.

For subsequent shots, his probability of scoring depends on whether he made the previous shot or not:

If he scored shot $i-1$, the probability of scoring shot i is $\frac{p_1}{q_1}$.

If he missed shot $i-1$, the probability of scoring shot i is $\frac{p_2}{q_2}$.

Da7doo7 is asking for your help to find the expected value of the number of balls that will score.

We can show that the answer can be written in the form $\frac{P}{Q}$ where P, Q are coprime integers and $Q \neq 0$ mod 998244353. Output the value of $(P \cdot Q^{-1})$ modulo 998244353.

Input

The first line contains a single integer T $(1 \leq T \leq 5 \times 10^4)$ — the number of testcases.

The first line of each testcase contains an integer n $(1 \leq n \leq 10^{12})$ — the number of balls he will shoot.

The second line of each testcase contains six integers $p_0, q_0, p_1, q_1, p_2, q_2$ $(0 \leq p_0, p_1, p_2 \leq 10^3$, $1 \leq q_0, q_1, q_2 \leq 10^3)$.

Output

An integer representing the expected value of the number of balls that will score, modulo 998244353.

Example

standard input	standard output
5 2 5 10 2 10 3 10 10 7 10 5 10 2 10 5 1 5 7 10 9 10 100 1 1 1 10 1 1 100000000 75 100 55 100 23 100	249561089 965715715 403450441 210477862 330198862

Problem H. How Ali Sees Black

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark Blue`

Ali wanted to make a very hard problem, so he thought really hard and after a week of continuous work, he came up with this problem:

You have a set M containing all numbers from 1 to n. In one operation, you can choose an integer x and subtract it from all numbers that are greater than or equal to it. If a number becomes zero, you remove it from the set. What is the minimum number of operations required to make the set empty?

Input

The only line of input contains one integer n $(1 \leq n \leq 100)$,as described in the problem.

Output

Print one integer, the minimum number of operations required to make the set empty.

Examples

standard input	standard output
1	1
2	2
47	6

Note

In the first example, the set is $\{1\}$ so we choose $x = 1$, and after this operation, the set will be empty.

In the second example, the set is: $\{1, 2\}$, we can choose $x = 2$ so the set will become: $\{1\}$,and after that we choose $x = 1$.

Problem I. Bored Kifah

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Black`

In a rectangular room, bored *Kifah* lives. *Kifah* used to walk scanning the whole room in a specific pattern while observing the digits written on the tiles of the room. At that moment, he wondered about the following imaginary problem:

Imagine a rectangular room with dimensions $n \times m$. You scan the room using the following pattern:

- Scan the first line with length m from the left to the right.
- Scan the second line with length m from the right to the left.
- Scan the third line with length m from the left to the right.
- And so on.

On each tile of the $n \times m$ tiles, one digit is written. Imagine that there are some determined rectangles that cover some of the room. All the rectangles are inside the room and they may overlap. The borders of the rectangles and the borders of the room are parallel to each other. You have a sheet of paper and an empty basket. At the beginning, the sheet of paper is considered empty but if you want to answer what is the number written on the paper, you can say it's zero. While scanning the room using the previous pattern, you will do the following:

- If the current tile is covered by at least one of the rectangles, you do one of the following:
  - Do not do any thing.
  - Add the digit of the current tile to the left of the number written on the paper (if the paper is empty, just put the number at the middle of the paper).
  - Add the digit of the current tile to the right of the number written on the paper (if the paper is empty, just put the number at the middle of the paper).
- If the current tile is not covered by any of the rectangles, you put the paper in the basket and continue with a new empty paper (even if the paper is empty, you put it in the basket and consider that it has a number of zero).

At the end of your walk, you look inside the basket. If there is at least one paper in the basket with a number written on it (the number that consists of the digits you added to the paper while walking) within the range $[A, B]$, we consider the walk as a **good walk**. We consider two walks different if the decision made at some tile in the first walk differs from the decision made at the same tile in the second walk even if both of the decisions have led to the same result. The two numbers A and B are your favourite numbers so you know them previously.

You will be asked multiple queries, for each query, you will be given a set of rectangles and asked to answer the following question: For all the possible walks that you can make, what is the number of the good walks.

Input

The first line contains two integers n, m ($1 \le n, m \le 20$) — the dimension of the room.

Then you will read a grid $a_{i,j}$ of n lines and m columns ($1 \le i \le n; 1 \le j \le m$). Where ($1 \le a_{i,j} \le 9$). — the digits written on the tiles of the room.

The next line has two integers A, B $(1 \le A, B \le 10^{18})$ — your favourite numbers.

The next line contains one integer q $(1 \le q \le 1000)$ — the number of queries.

For each query, the first line is s $(1 \le s \le 1000)$ — the number of rectangles at this query.

For the following s lines, each line contains four numbers x_1, y_1, x_2, y_2 $(1 \le x_1 \le x_2 \le n; 1 \le y_1 \le y_2 \le m)$ — the upper left corner of the rectangle (x_1, y_1) and the lower right corner of the rectangle (x_2, y_2) respectively.

Output

For each query, print the number of good walks considering the given rectangles in that query. Output the answer modulo $10^9 + 7$.

Examples

standard input	standard output
2 3 5 3 1 3 9 2 2 30 2 1 1 1 1 1 1 1 3 2 3	2 6
3 4 3 6 3 9 1 3 4 2 7 1 8 9 525 4487960024 1 6 2 4 2 4 2 1 2 2 1 3 2 3 2 4 3 4 2 4 2 4 1 1 1 2	1686

Note

For the first query there are 3 different walks:

- you don't take 5, getting 0.
- you add 5 to the left, getting 5.
- you add 5 to the right, getting 5.

For the second query there are 9 different walks:

- you don't take 1 then don't take 2, getting 0.
- you don't take 1 then add 2 to the left, getting 2.
- you don't take 1 then add 2 to the right, getting 2.

- you add 1 to the left then don't take 2, getting 1.
- you add 1 to the left then add 2 to the left, getting 21.
- you add 1 to the left then add 2 to the right, getting 12.
- you add 1 to the right then don't take 2, getting 1.
- you add 1 to the right then add 2 to the left, getting 21.
- you add 1 to the right then add 2 to the right, getting 12.

Problem J. Short Statement

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Light Blue`

The setter of this problem hates long statements' problems, so the problem is:

The **Score** of array b of size m equals:

$$\sum_{i=1}^{m-1} gcd(b_i, b_{i+1})$$

You are given an array a of n integers and an integer k.

You need to find a non-empty subsequence a_{i_1}, a_{i_2}, ..., a_{i_m} with **maximum Score** such that:

$i_j < i_{j+1}$ and i_{j+1} - $i_j \le k$ for all $1 \le j \le m - 1$.

Print the **Score** of this subsequence.

Input

The first line contains a single integer $t(1 \le t \le 100)$ — the number of testcases.

Then the descriptions of t testcases follow.

The first line of each testcase contains two integers n and k— the size of the array a $(2 \le n \le 2 \cdot 10^5)$ and k explained above $(1 \le k \le n - 1)$.

The second line consists of n integers a_1, a_2, ..., a_n $(1 \le a_i \le 4 \cdot 10^5)$. It's guaranteed that the sum of n over all testcases won't exceed $2 \cdot 10^5$.

Output

Output for each testcase the maximum **Score**.

Example

standard input	standard output
2 5 1 2 4 8 16 32 5 2 2 4 1 16 32	30 22

Problem K. Bad Friend

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Red`

In an imaginary country called IM, there are n cities and m **directed** roads. However, these roads may not connect all the cities.

Kifah has been a resident of IM for many years and knows the country well. On a sunny morning, while chatting with his friend *Abdullah*, a claim was made. *Abdullah* stated that he had moved to IM but was unsure of his exact location. Skeptical, *Kifah* didn't believe him and demanded proof. In response, *Abdullah* provided a set of city pairs, asserting that for each pair, there exists a path starting from the first city and ending at the second, passing through *Abdullah's* current city.

Can you help *Kifah* to check if there is any contradiction in *Abdullah's* claims.

In this Problem you will be given a graph consists of n nodes and m **directed** edges. Note that the graph may **not** be connected, but it is guaranteed that there is no multi-edges or self-loops.

Then you will be given a set of q pairs of nodes (a, b) that claims: there is a path starts from a, ends at b and passes by the city where *Abdullah* is. These paths may traverse nodes or roads multiple times.

Your goal is to help *Kifah* determine whether *Abdullah's* claims hold without contradiction, thus verifying his honesty.

Input

The first line contains a single integer Tc $(1 \leq Tc \leq 5 \cdot 10^4)$ — the number of test cases.

The first line of each test case contains two integers n, m $(1 \leq n \leq 5 \cdot 10^4,\ 1 \leq m \leq min(n \cdot (n-1), 10^5))$ — the number of vertices and edges in the graph respectively.

For the following m lines, each line contains two integers (u, v) $(1 \leq u, \text{v} \leq n)$, describing a directed edge from vertex u to vertex v. It is guaranteed that there are no multi-edges or self-loops in the graph.

The next line contains a single integer q $(1 \leq q \leq 5 \cdot 10^4)$, the number of pairs *Abdullah* provided.

For the following q lines, each line contains two integers (a, b) $(1 \leq a, \text{b} \leq n)$, describing a claim that says: There is a path starts from a, ends at b and pass by the city where *Abdullah* is. These paths may traverse nodes or roads multiple times.

It is guaranteed that both the sum of n and q over all test cases doesn't exceed $5 \cdot 10^4$.

It is guaranteed that the sum of m over all test cases doesn't exceed 10^5.

Output

For each test case, output "YES"(without quotes) if *Kifah* can't find any contradiction in *Abdullah's* claims, and "NO"(without quotes) otherwise.

Example

standard input	standard output
2 4 3 1 2 2 3 3 4 2 1 3 2 4 5 4 1 2 2 3 4 3 5 3 2 1 3 5 4	YES NO

Note

In the first testcase, both nodes 2 and 3 meet the claims. For node 2 the path $1 \to 2 \to 3$ meets the first pair $(1, 3)$, and the path $2 \to 3 \to 4$ meets the second pair $(2, 4)$. So we can't say that *Abdullah* is a Liar, that's why we provide "YES".

In the second testcase, there is no node that meets the constraints. For the first pair in the constraints $(1, 3)$ the only path that meets this constraint is $1 \to 2 \to 3$. On the other hand, for the second pair in the constraints $(5, 4)$, there is no path that can start from 5 and end at 4. So we can say that *Abdullah* is a Liar because there is a contradiction, that's why we provide "NO".

Problem L. Reckless Driving

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`White`

A truck driver was asked to transport $2n$ boxes of active chemical substances. He is in a hurry and does not care about bumps in the road that will cause the substances in the back to react with each other. Initially, there were n boxes of type A and n boxes of type B placed over each other carefully in a certain arrangement forming a tower of height $2n$.

The truck jumps upon hitting a road bump, and all the boxes fly in the air. A random box (uniformly picked from boxes in the air) hits the truck ground first, then another random box (also picked uniformly) hits the first box, and so on. Only if an active box hits another active box beneath it of the opposite type (A hits B or B hits A), they react with each other, releasing a gas C, and both boxes become inactive. The clever driver smells the gases after each bump and stops to get rid of the inactive boxes, then places the remaining active boxes over each other with any order he wants carefully (without causing reactions) and continues driving recklessly!

What is the expected number of bumps after which the truck becomes empty, and the driver gets fired?

It can be shown that this answer can always be expressed in the form P/Q where P, Q are coprime integers and $Q \neq 0 \bmod 998244353$. Output the value of $(P \cdot Q^{-1})$ modulo 998244353.

Input

The first line contains a single integer $1 \leq T \leq 20$ — the number of test cases.

Each of the following t lines contains an integer $0 \leq n \leq 200$ — both the number of A boxes and the number of B boxes.

Output

For each of the testcases, print the expected number of bumps modulo 998244353.

Example

standard input	standard output
4 0 1 2 3	0 1 332748119 232923684

Note

For the case $n = 3$, one way the process can happen:

After the first bump, the boxes get shuffled randomly, and the reaction happens from left to right (down to up in the problem statement) as follows:

ABAABB -> [AB]AABB -> [AB]A[AB]B -> [C]A[C]B

The driver then gets rid of the reactants. When the second bump comes, the remaining boxes get shuffled randomly and react again, for example:

BA -> [BA] -> [C]

Finally, the driver gets rid of the reactants, and the truck becomes empty; here we needed 2 bumps.

Problem M. Ali and BOX

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark Green`

Ali is playing a very boring game called *BOX*. In this game, you have a box that contains x coins, and there is a road which contains n checkpoints where you can get or lose coins. The road can be represented as an array a containing n integers where a_i represents the amount of coins that you may gain or lose at the ith checkpoint. You start at the beginning of the road "at index 0". The game goes as follows:

You start at index zero with an initial amount of coins equals to x. You start visiting the checkpoints from left to right without skipping any of them and with the capability of ending the game any time you want.

At each checkpoint, you get or lose an amount of coins defined by a_i. If at any point, the total amount of coins you have becomes negative, you lose the game.

You can do this operation **only for once**: pay c coins "if you have them"and choose an index i where i is after your current checkpoint, then reverse the subsegment $[i, n]$.

The goal of the game is to maximize the amount of coins you have. Ali isn't very good at this kind of game (he only plays shooters) so he asked you to help him. Can you find the maximum amount of coins Ali can get?

Input

The first line contains three integers: n, x, c $(1 \leq n \leq 10^6), (0 \leq x, c \leq 10^9)$, number of checkpoints, the initial value of coins in the box, and the cost of flipping a segment.

The second line contains n integers, $a_1, a_2, ..., a_n$ $(-10^9 \leq a_i \leq 10^9)$

Output

Print one integer, the maximum amount of coins Ali can get.

Examples

standard input	standard output
4 0 1 1 -2 3 4	7
13 19 22 12 30 -1 15 -55 -9 17 5 -4 6 -15 10 15	87

Note

In the first example, we move to the first checkpoint and add it to x, so now $x = 1$, then we pay $c = 1$ and flip the subsegment $[2, 4]$, so the array becomes: $[1\ 4\ 3\ -2]$, now we move to the second and the third checkpoints and then we end the game, so the final result will be: $x = 0 + 1 - 1 + 4 + 3 = 7$.