

## Задача А. Напишіть чекер

Ліміт часу: 1 second

Ліміт використання пам'яті: 256 megabytes

Організатори хотіли зробити цю задачу дуже простою, просто  $A+B$ . Вам мали бути дані два числа  $a$  і  $b$ , і ви повинні були б вивести їхню суму  $a + b$ .

Але організатори були так зайняті підготовкою цього змагання, що забули написати чекер для цієї задачі. Будь ласка, допоможіть їм: який вердикт має отримати дана відповідь: **Accepted**, чи **Wrong Answer**?

### Формат вхідних даних

Перший рядок вводу містить два цілі числа  $a, b$  ( $1 \leq a, b \leq 10^9$ ).

Другий рядок вводу містить одне ціле число  $answer$  ( $1 \leq answer \leq 2 \cdot 10^9$ ) — відповідь, яку ви повинні перевірити.

### Формат вихідних даних

Якщо  $answer = a + b$ , виведіть **Accepted**. В іншому випадку виведіть **Wrong Answer**.

### Приклади

standard input	standard output
2 2 4	Accepted
42 69 100	Wrong Answer

### Зауваження

У першому прикладі, оскільки  $2 + 2 = 4$ , відповідь правильна, і ви повинні вивести **Accepted**.

У другому прикладі, оскільки  $42 + 69 = 111 \neq 100$ , відповідь неправильна, і ви повинні вивести **Wrong Answer**.

## Задача В. Фіксовані точки

Ліміт часу: 1 second

Ліміт використання пам'яті: 256 megabytes

Для масиву  $a$  визначимо його вагу як кількість елементів, які не зміняться після сортування. Наприклад, вага масиву  $[7, 3, 1, 10]$  дорівнює 2: після сортування він стане рівним  $[1, 3, 7, 10]$ , тому другий і четвертий елементи не зміняться, а інші два зміняться.

Вам дано перестановку  $p$  цілих чисел від 1 до  $n$ . Знайдіть суму ваг всіх її  $\frac{n(n+1)}{2}$  підмасивів.

Тут під підмасивом мається на увазі послідовний відрізок масиву. Наприклад, масив  $[7, 3, 1, 10]$  має 10 підмасивів:  $[7]$ ,  $[3]$ ,  $[1]$ ,  $[10]$ ,  $[7, 3]$ ,  $[3, 1]$ ,  $[1, 10]$ ,  $[7, 3, 1]$ ,  $[3, 1, 10]$ ,  $[7, 3, 1, 10]$ .

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^4$ ) — кількість тестових наборів. Далі йде опис тестових наборів.

Перший рядок кожного тестового набору містить одне ціле число  $n$  ( $1 \leq n \leq 2000$ ) — довжину перестановки.

Другий рядок кожного тестового набору містить  $n$  цілих чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ , всі  $p_i$  попарно різні) — елементи перестановки.

Гарантується, що сума  $n^2$  по всіх тестових наборах не перевищує  $4 \cdot 10^6$ .

### Формат вихідних даних

Для кожного тесту виведіть одне ціле число — суму ваг всіх його  $\frac{n(n+1)}{2}$  підмасивів.

### Приклад

standard input	standard output
3	20
4	9
1 2 3 4	37
5	
5 4 3 2 1	
8	
4 6 1 2 8 5 3 7	

### Зауваження

У першому тестовому наборі для кожного підмасиву всі його елементи після сортування залишаться незмінними. Тому відповідь дорівнює  $1 \cdot 4 + 2 \cdot 3 + 3 \cdot 2 + 4 \cdot 1 = 20$ , оскільки є 4 підмасиви довжиною 1, 3 підмасиви довжиною 2, 2 підмасиви довжиною 3 і 1 підмасив довжиною 4.

## Задача С. Залишкові конфігурації

Ліміт часу: 1 second

Ліміт використання пам'яті: 256 megabytes

Вам дано масив  $a_1, a_2, \dots, a_n$  додатних цілих чисел. Ви можете виконувати наступну операцію будь-яку кількість разів:

- Виберіть додатне ціле число  $x$ . Потім замініть  $a_i$  на  $a_i \bmod x$  для кожного  $1 \leq i \leq n$ .

Скільки різних масивів  $a_1, a_2, \dots, a_n$  можна отримати за допомогою цих операцій? Оскільки ця кількість може бути дуже великою, виведіть її за модулем 998244353.

Тут  $x \bmod y$  позначає залишок від ділення  $x$  на  $y$ . Наприклад,  $6 \bmod 3 = 0$ , а  $6 \bmod 4 = 2$ .

### Формат вхідних даних

Перший рядок вводу містить одне ціле число  $n$  ( $1 \leq n \leq 500$ ) — довжину масиву.

Другий рядок вводу містить  $n$  цілих чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 500$ ) — елементи масиву.

### Формат вихідних даних

Виведіть одне ціле число — кількість різних масивів, які можна отримати за допомогою цих операцій.

### Приклади

standard input	standard output
1 5	4
2 6 5	7
5 1 2 4 8 16	69

### Зауваження

У першому прикладі ви можете отримати наступні масиви:  $[5]$ ,  $[2]$ ,  $[1]$ ,  $[0]$ .

У другому прикладі ви можете отримати наступні масиви:  $[6, 5]$ ,  $[2, 1]$ ,  $[1, 0]$ ,  $[0, 5]$ ,  $[0, 2]$ ,  $[0, 1]$ ,  $[0, 0]$ .

## Задача D. XOR > AND

Ліміт часу: 1 second

Ліміт використання пам'яті: 256 megabytes

Вам дано  $n$  чисел  $a_1, a_2, \dots, a_n$ . Чи можете ви розбити їх на дві **непорожні** групи так, щоб XOR чисел у першій групі був більший за AND чисел у другій групі?

Нагадаємо, що XOR позначає операцію **побітового виключного АБО**, а AND позначає операцію **побітового І**. Наприклад:

- $13 \text{ XOR } 6 = 11$ , адже в двійковому записі  $13 = 1101$ , а  $6 = 0110$ , тому їх XOR має бути рівним  $1011 = 11$ ;
- $13 \text{ AND } 6 = 4$ , адже в двійковому записі  $13 = 1101$ , а  $6 = 0110$ , тому їх AND має бути рівним  $0100 = 4$ .

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^5$ ) — кількість тестових наборів. Далі слідує опис тестових наборів.

Перший рядок кожного тестового набору містить одне ціле число  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) — кількість чисел.

Другий рядок кожного тестового набору містить  $n$  цілих чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i < 2^{30}$ ).

Гарантується, що сума  $n$  по всіх тестових наборах не перевищує  $2 \cdot 10^5$ .

### Формат вихідних даних

Для кожного тестового набору, якщо немає способу розбити числа, що задовольняє умові, виведіть NO.

Інакше, виведіть YES. На наступному рядку виведіть одне ціле число  $k$  ( $1 \leq k \leq n - 1$ ) — розмір першої групи.

На наступному рядку виведіть  $k$  цілих чисел  $pos_1, pos_2, \dots, pos_k$  ( $1 \leq pos_i \leq n$ , всі  $pos_i$  попарно різні) — **позиції** елементів, які ви збираєтесь помістити в першу групу.

Ви можете вивести YES і NO у будь-якому регістрі (наприклад, рядки `yEs`, `yes`, `Yes` будуть вважатися позитивною відповіддю).

### Приклад

standard input	standard output
3	YES
2	1
3 2	1
4	NO
69 69 69 69	YES
5	3
8 9 10 11 12	3 4 5

### Зауваження

У першому тестовому наборі ми можемо помістити число 3 в першу групу, а число 2 в другу групу. XOR чисел у першій групі буде 3, а AND чисел у другій групі буде 2.

У другому тестовому наборі немає способу розбити числа.

У третьому тестовому наборі ми можемо помістити числа 10, 11, 12 в першу групу, а числа 8, 9 в другу групу. XOR чисел у першій групі буде  $10 \text{ XOR } 11 \text{ XOR } 12 = 13$ , а AND чисел у другій групі буде  $8 \text{ AND } 9 = 8$ .

## Задача Е. Неспадний

Ліміт часу: 1 second  
Ліміт використання пам'яті: 256 megabytes

Вам дано масив цілих чисел  $a$  довжиною  $n$  та ціле число  $x$ . Ви можете виконувати наступну операцію:

- Оберіть будь-яке  $i$  таке, що  $1 \leq i \leq n$ , і встановіть  $a_i = x$ .

Знайдіть найменшу кількість операцій, яку вам потрібно виконати, щоб зробити масив  $a$  неспадним. Масив  $a$  довжиною  $n$  називається неспадним, якщо  $a_1 \leq a_2 \leq \dots \leq a_n$ .

Оскільки ця задача занадто проста, вирішіть її для кожного  $x$  від 1 до  $n$ .

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^4$ ) — кількість тестових наборів. Далі слідує опис тестових наборів.

Перший рядок кожного тестового набору містить одне ціле число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — довжину масиву.

Другий рядок кожного тестового набору містить  $n$  цілих чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — елементи масиву.

Гарантується, що сума  $n$  по всіх тестових наборах не перевищує  $2 \cdot 10^5$ .

### Формат вихідних даних

Для кожного тестового набору виведіть  $n$  цілих чисел — відповіді на задачу для  $x = 1, 2, \dots, n$ .

### Приклад

standard input	standard output
4	2 2 2 3
4	0 0 0 0
2 1 3 2	1 1 3 3 4
4	3 4 3 4 3 4 3 4
1 2 2 4	
5	
3 2 2 2 4	
8	
1 3 5 7 1 3 5 7	

### Зауваження

Для першого тестового набору нижче наведено оптимальні операції для  $x = 1, 2, 3, 4$ :

- Для  $x = 1$ : ми можемо перетворити  $[2, 1, 3, 2]$  в  $[1, 1, 1, 2]$  замінивши перший і третій елементи на 1.
- Для  $x = 2$ : ми можемо перетворити  $[2, 1, 3, 2]$  в  $[2, 2, 2, 2]$  замінивши другий і третій елементи на 2.
- Для  $x = 3$ : ми можемо перетворити  $[2, 1, 3, 2]$  в  $[2, 3, 3, 3]$  замінивши другий і четвертий елементи на 3.
- Для  $x = 4$ : ми можемо перетворити  $[2, 1, 3, 2]$  в  $[2, 4, 4, 4]$  замінивши останні три елементи на 4.

## Задача F. Антипаліндромний

Ліміт часу: 1 second

Ліміт використання пам'яті: 256 megabytes

Вам дано рядок  $s$ . Ви хочете переставити його символи, отримуючи рядок  $r$ , так щоб виконувалася наступна умова:

- Для кожного  $i = 2, 3, \dots, |s|$ , рядок  $r_1 r_2 \dots r_i$  **не** є паліндромом.

Чи можете ви це зробити? Якщо таких рядків  $r$  декілька, ви можете вивести довільний з них.

Нагадаємо, що рядок  $s$  називається **паліндромом**, якщо він читається однаково зліва направо і справа наліво. Наприклад, `tenet` є паліндромом.

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^5$ ) — кількість тестових наборів. Далі йде опис тестових наборів.

Єдиний рядок кожного тестового набору містить рядок  $s$ , що складається з малих латинських літер.

Гарантується, що сума довжин  $s$  по всіх тестових наборах не перевищує  $10^6$ .

### Формат вихідних даних

Для кожного тестового набору, якщо немає потрібного способу переставити символи  $s$ , виведіть NO.

В іншому випадку, виведіть YES. На наступному рядку виведіть рядок  $r$ . Він повинен бути перестановкою символів  $s$  і задовольняти вимогам з умови.

Ви можете виводити YES і NO в будь-якому регістрі (наприклад, рядки `yEs`, `yes`, `Yes` будуть вважатися позитивною відповіддю).

### Приклад

standard input	standard output
5	YES
a	a
sos	YES
abba	oss
icpc	NO
tenet	YES
	icpc
	YES
	tente

## Задача G. Конфігурації мінімумів

Ліміт часу: 3 seconds

Ліміт використання пам'яті: 256 megabytes

Задано два цілі числа  $n$  та  $m$ . Знайдіть кількість пар  $(a, b)$  цілих масивів довжиною  $n$ , для яких виконуються наступні умови:

- Усі елементи  $a$  та  $b$  є цілими числами від 1 до  $m$ ;
- Для будь-яких  $1 \leq i < j \leq n$ ,  $\min(a_i, b_j) = \min(a_j, b_i)$ .

Оскільки ця кількість може бути великою, виведіть її за модулем 998244353.

### Формат вхідних даних

У єдиному рядку вводу містяться два цілі числа  $n, m$  ( $1 \leq n, m \leq 10^6$ ).

### Формат вихідних даних

Виведіть одне ціле число — відповідь до задачі за модулем 998244353.

### Приклади

standard input	standard output
1 3	9
2 2	10
69 42	608932821

### Зауваження

У першому прикладі будь-яка пара масивів  $([x], [y])$  з  $1 \leq x, y \leq 3$  задовольняє умовам задачі, їх 9.

У другому прикладі ось відповідні пари масивів:

- $([1, 1], [1, 1])$ ;
- $([1, 1], [1, 2])$ ;
- $([1, 1], [2, 1])$ ;
- $([1, 1], [2, 2])$ ;
- $([1, 2], [1, 1])$ ;
- $([1, 2], [1, 2])$ ;
- $([2, 1], [1, 1])$ ;
- $([2, 1], [2, 1])$ ;
- $([2, 2], [1, 1])$ ;
- $([2, 2], [2, 2])$ .

## Задача Н. Різні суми

Ліміт часу: 1 second

Ліміт використання пам'яті: 256 megabytes

Дано масив цілих чисел  $a$  довжиною  $n$ . Ви хочете розбити його на найбільшу можливу кількість послідовних підмасивів, так щоб виконувалася наступна умова:

- Нехай  $k$  — кількість підмасивів, а  $s_1, s_2, \dots, s_k$  — суми елементів у відповідних підмасивах. Тоді  $s_i \neq s_{i+1}$  для всіх  $1 \leq i \leq k - 1$ . (Зверніть увагу, що вам не потрібно, щоб всі  $k$  сум були попарно різними, вам потрібно лише, щоб сусідні суми були різними).

На яку найбільшу кількість підмасивів ви можете розбити масив  $a$  за такої умови?

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^5$ ) — кількість тестових наборів. Далі йде опис кожного тестового випадку.

Перший рядок кожного тестового набору містить одне ціле число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — довжину масиву.

Другий рядок кожного тестового набору містить  $n$  цілих чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — елементи масиву.

### Формат вихідних даних

Для кожного тестового набору виведіть одне ціле число — відповідь на задачу.

### Приклад

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

### Зауваження

У першому тестовому наборі ми можемо розбити  $[1, 2, 3, 4]$  на 4 підмасиви  $[1], [2], [3], [4]$ . Суми в цих підмасивах дорівнюють 1, 2, 3, 4, жодні дві сусідні суми не рівні.

У другому тестовому наборі ми можемо розбити  $[1, 1, 1, 1]$  на 3 підмасиви  $[1], [1, 1, 1], [1]$ . Суми в цих підмасивах дорівнюють 1, 3, 1, жодні дві сусідні суми не рівні.

У третьому тестовому наборі ми можемо розбити  $[2, 1, 1, 1, 1, 2]$  на 5 підмасивів  $[2], [1], [1, 1], [1], [2]$ . Суми в цих підмасивах дорівнюють 2, 1, 2, 1, 2, жодні дві сусідні суми не рівні.



## Задача I. Максимальні суми підмасивів

Ліміт часу: 3 seconds

Ліміт використання пам'яті: 256 megabytes

Для масиву цілих чисел  $a$  визначимо  $f(a)$  як найбільшу суму елементів будь-якого з його послідовних підмасивів (можливо, порожнього). Наприклад,  $f([-1, -2, -1]) = 0$ , а  $f([1, -2, 3, 1, -2, 1])$  дорівнює 4, що досягається на  $[3, 1]$ .

Також, для двох масивів  $a, b$  визначимо  $concat(a, b)$  як їхню конкатенацію. Наприклад,  $concat([1, 3], [2, 4]) = [1, 3, 2, 4]$ .

Вам надано  $n$  масивів  $arr_1, arr_2, \dots, arr_n$ . Знайдіть суму  $f(concat(arr_i, arr_j))$  по всіх парах  $(i, j)$  цілих чисел з  $1 \leq i, j \leq n$ .

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^4$ ) — кількість тестових наборів. Далі йде опис тестових наборів.

Перший рядок кожного тестового набору містить одне ціле число  $n$  ( $1 \leq n \leq 4 \cdot 10^5$ ) — кількість масивів.

$i$ -й з наступних  $n$  рядків містить ціле число  $k_i$  та  $k_i$  цілих чисел  $arr_{i,1}, arr_{i,2}, \dots, arr_{i,k_i}$  ( $1 \leq k_i \leq 4 \cdot 10^5$ ,  $-10^7 \leq arr_{i,j} \leq 10^7$ ) — довжину  $arr_i$  та його елементи.

Гарантується, що сума довжин усіх масивів у всіх тестових наборах не перевищує  $4 \cdot 10^5$ .

### Формат вихідних даних

Для кожного тестового набору виведіть одне ціле число — суму  $f(concat(arr_i, arr_j))$  по всіх парах  $(i, j)$  цілих чисел з  $1 \leq i, j \leq n$ .

### Приклад

standard input	standard output
3	20
2	22
3 1 1 -10	131
2 2 2	
3	
3 -10 1 -10	
3 -10 2 -10	
3 -10 3 -10	
5	
1 1	
2 5 -5	
3 -2 1 2	
5 3 -10 5 -10 3	
5 1 -10 5 -10 3	

### Зауваження

У першому тестовому наборі необхідно знайти суму  $f$  наступних 4 підмасивів:  $f([1, 1, -10, 1, 1, -10]) = 2$ ,  $f([1, 1, -10, 2, 2]) = 4$ ,  $f([2, 2, 1, 1, -10]) = 6$ ,  $f([2, 2, 2, 2]) = 8$ .

Загальна сума дорівнює  $2 + 4 + 6 + 8 = 20$ .

## Задача J. Максимізуйте префіксні максимуми

Ліміт часу: 1 second

Ліміт використання пам'яті: 256 megabytes

Для масиву  $a$  довжиною  $n$  визначимо  $f(a)$  як кількість позицій  $1 \leq i \leq n$ , для яких  $a_i = \max(a_1, a_2, \dots, a_i)$ . Іншими словами,  $f(a)$  — кількість префіксних максимумів в масиві  $a$ .

Вам задана перестановка  $(p_1, p_2, \dots, p_n)$  цілих чисел від 1 до  $n$ . Знайдіть будь-яку перестановку  $(q_1, q_2, \dots, q_n)$  цілих чисел від 1 до  $n$ , для якої виконується наступна умова:

- $f((q_1, q_2, \dots, q_n)) + f((p_{q_1}, p_{q_2}, \dots, p_{q_n}))$  є **найбільшим** серед усіх перестановок  $(q_1, q_2, \dots, q_n)$ .

Якщо існує декілька таких перестановок, знайдіть будь-яку з них.

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^5$ ) — кількість тестових наборів. Далі слідує опис тестових наборів.

Перший рядок кожного тестового набору містить одне ціле число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — довжину перестановки.

Другий рядок кожного тестового набору містить  $n$  цілих чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ , всі  $p_i$  різні) — елементи перестановки.

Гарантується, що сума  $n$  по всіх тестових наборах не перевищує  $2 \cdot 10^5$ .

### Формат вихідних даних

Для кожного тестового набору виведіть  $n$  цілих чисел  $q_1, q_2, \dots, q_n$  ( $1 \leq q_i \leq n$ , всі  $q_i$  різні) — будь-яку перестановку цілих чисел від 1 до  $n$ , яка **максимізує** значення  $f((q_1, q_2, \dots, q_n)) + f((p_{q_1}, p_{q_2}, \dots, p_{q_n}))$

### Приклад

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

### Зауваження

У першому тестовому наборі, для  $q = (1, 2, 3)$ , значення дорівнює  $f(1, 2, 3) + f(p_1, p_2, p_3) = f(1, 2, 3) + f(1, 2, 3) = 3 + 3 = 6$ .

У другому тестовому наборі, для  $q = (1, 3, 4, 2)$ , значення дорівнює  $f(1, 3, 4, 2) + f(p_1, p_3, p_4, p_2) = f(1, 3, 4, 2) + f(2, 3, 1, 4) = 3 + 3 = 6$ .

У третьому тестовому наборі, для  $q = (1, 3, 5, 4, 2)$ , значення дорівнює  $f(1, 3, 5, 4, 2) + f(p_1, p_3, p_5, p_4, p_2) = f(1, 3, 5, 4, 2) + f(1, 2, 3, 4, 5) = 3 + 5 = 8$ .

## Задача К. Максимізуйте LCS перестановок

Ліміт часу: 2 seconds

Ліміт використання пам'яті: 256 megabytes

Дано перестановку  $(p_1, p_2, \dots, p_n)$  цілих чисел від 1 до  $n$ . Знайдіть будь-яку перестановку  $(q_1, q_2, \dots, q_n)$  цих чисел від 1 до  $n$  таку, що виконується наступна умова:

- $LCS((q_1, q_2, \dots, q_n), (p_{q_1}, p_{q_2}, \dots, p_{q_n}))$  є **найбільшою** серед усіх перестановок  $(q_1, q_2, \dots, q_n)$ .

Якщо існує декілька таких перестановок, знайдіть будь-яку з них.

Тут  $LCS(a, b)$  позначає довжину найбільшої спільної підпослідовності послідовностей  $a$  і  $b$ . Наприклад,  $LCS((1, 3, 4, 2, 5), (3, 1, 2, 4, 5)) = 3$ , а однією з спільних підпослідовностей довжини 3 є  $(1, 2, 5)$ .

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^5$ ) — кількість тестових наборів. Далі йде опис тестових наборів.

Перший рядок кожного тестового набору містить одне ціле число  $n$  ( $1 \leq n \leq 10^6$ ) — довжину перестановки.

Другий рядок кожного тестового набору містить  $n$  цілих чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ , всі  $p_i$  різні) — елементи перестановки.

Гарантується, що сума  $n$  по всіх тестових наборах не перевищує  $10^6$ .

### Формат вихідних даних

Для кожного тестового набору виведіть  $n$  цілих чисел  $q_1, q_2, \dots, q_n$  ( $1 \leq q_i \leq n$ , всі  $q_i$  різні) — будь-яку перестановку цілих чисел від 1 до  $n$ , яка **максимізує** значення  $LCS((q_1, q_2, \dots, q_n), (p_{q_1}, p_{q_2}, \dots, p_{q_n}))$ .

### Приклад

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

### Зауваження

У першому тестовому наборі, для  $q = (1, 2, 3, 4)$ , маємо:

$$LCS((1, 2, 3, 4), (p_1, p_2, p_3, p_4)) = LCS((1, 2, 3, 4), (1, 2, 3, 4)) = 4.$$

У другому тестовому наборі, для  $q = (1, 6, 2, 5, 3, 4)$ , маємо:

$$LCS((1, 6, 2, 5, 3, 4), (p_1, p_6, p_2, p_5, p_3, p_4)) = LCS((1, 6, 2, 5, 3, 4), (6, 1, 5, 2, 4, 3)) = 3; \text{ однією з спільних підпослідовностей довжиною 3 є } (1, 2, 3).$$

## Задача L. Не зростаючий

Ліміт часу: 1 second

Ліміт використання пам'яті: 256 megabytes

Вам дано масив  $a_1, a_2, \dots, a_n$  з додатніми цілими числами. За одну операцію ви можете збільшити будь-який елемент масиву на 1.

Ви ненавидите зростаючі масиви. Знайдіть найменшу кількість операцій, необхідних, щоб зробити масив **не** зростаючим.

Масив називається зростаючим, якщо  $a_1 < a_2 < \dots < a_n$ .

### Формат вхідних даних

Перший рядок містить одне ціле число  $t$  ( $1 \leq t \leq 10^4$ ) — кількість тестових наборів. Далі слідує описи тестових наборів.

Перший рядок кожного тестового набору містить одне ціле число  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) — довжину масиву.

Другий рядок кожного тестового набору містить  $n$  цілих чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — елементи масиву.

Гарантується, що сума  $n$  по всіх тестових наборах не перевищує  $2 \cdot 10^5$ .

### Формат вихідних даних

Для кожного тестового набору виведіть одне ціле число — найменшу кількість операцій, необхідних, щоб зробити масив не зростаючим.

### Приклад

standard input	standard output
3	27
2	0
42 69	111
3	
1 1 1	
5	
123 234 345 456 567	

### Зауваження

У першому тестовому наборі початковий масив є зростаючим. Один зі способів зробити його не зростаючим за 27 операцій — 27 разів збільшити перший елемент на 1. Масив стане рівним [69, 69].

У другому тестовому наборі початковий масив вже не зростаючий.

У третьому тестовому наборі початковий масив є зростаючим. Один зі способів зробити його не зростаючим за 111 операцій — 111 разів збільшити другий елемент на 1. Масив стане рівним [123, 345, 345, 456, 567].