

## Wild operations (wild)

Валентин хоче перевірити здатність Романа виконувати складні операції з масивами, тому він дав йому масив  $A_0, \dots, A_{N-1}$  довжини  $N$ .

Тепер Валентин попросить Романа виконати деякі операції з масивом, де кожна операція може бути або:

- *змінити* значення  $A_p$  на  $x$ , для деякого цілого  $x$  і коректного індексу  $p$ .
- *збурити* діапазон  $[l, r]$ , тобто встановити  $A_p = \max(A_p, A_{p-1})$  **одночасно** для всіх  $l < p \leq r$ .

У будь-який момент Валентин може попросити Романа сказати йому значення  $A_p$  для деякого коректного індексу  $p$ .

Роман дуже зайнятий, тому він вирішив попросити Вашої допомоги відповідати на запитання Валентина.

### Реалізація

Ви повинні надіслати єдиний файл з розширенням `.cpp`.

← Серед вкладень до цієї задачі Ви знайдете шаблон `wild.cpp` з прикладом реалізації.

Ви повинні реалізувати такі функції:

C++	<code>void init(int N, vector&lt;int&gt; A);</code>
-----	---

- Ця функція викликається один раз на початку виконання програми.
- Ціле число  $N$  — довжина масиву.
- Масив  $A$ , проіндексований від 0 до  $N - 1$ , — початковий масив, який вибрав Валентин.

C++	<code>void change(int p, int x);</code>
-----	---

- Ця функція викликається багато разів під час виконання програми, коли Валентин робить зміну.
- Ціле число  $p$  — індекс зміненого значення.
- Ціле число  $x$  — нове значення для призначення.

C++	<code>void perturb(int l, int r);</code>
-----	--

- Ця функція викликається багато разів під час виконання програми, коли Валентин збурює діапазон.
- Ціле число  $l$  — ліва межа діапазону.
- Ціле число  $r$  — права межа діапазону.

C++	<code>int calc(int p);</code>
-----	-------------------------------

- Ця функція викликається багато разів під час виконання програми, коли Валентин запитує значення елемента.

- Ціле число  $p$  — індекс елемента, який запитує Валентин.
- Функція повинна повертати значення  $A_p$  після виконання всіх попередніх операцій.

## Приклад градера

Спрощена версія градера, що використовувався під час перевірки, доступна в каталозі задачі. Ви можете використати його для локального тестування. Цей градер читає вхідні дані з `stdin`, викликає потрібну функцію та записує результат у `stdout` у такому форматі.

Нехай  $Q$  — загальна кількість змін, збурень і запитань від Валентина.

Тоді вхідний файл складається з  $2 + Q$  рядків, що містять:

- Рядок 1: цілі числа  $N, Q$ .
- Рядок 2:  $N$  цілих чисел  $A_0, \dots, A_{N-1}$ , початкові значення масиву.
- Рядок  $3 + i$  ( $0 \leq i < Q$ ): 2 або 3 числа, в одному з форматів:
  - $1 \ p \ x$ : Валентин змінює  $A_p$  на  $x$ .
  - $2 \ l \ r$ : Валентин збурює діапазон  $[l, r]$ ;
  - $3 \ p$ : Валентин запитує значення  $A_p$ .

Вихідний файл складається з  $Q_3$  рядків (де  $Q_3$  — кількість викликів `calc`), що містять значення, які повертає функція `calc`.

## Обмеження

- $1 \leq N \leq 400\,000$ .
- $0 \leq Q \leq 400\,000$ .
- $1 \leq A_i \leq 10^9$  для всіх  $0 \leq i < N$ .
- $0 \leq p < N$  у кожному виклику `change` і `calc`.
- $0 \leq l < r \leq N - 1$  у кожному виклику `perturb`.
- $1 \leq x \leq 10^9$  у кожному виклику `change`.

## Оцінювання

Вашу програму буде протестовано на кількох тестових наборах, згрупованих у підзадачі. Щоб отримати бали за підзадачу, потрібно правильно розв'язати всі тести в ній.

Нехай  $Q_1$  — кількість викликів функції `change` у тесті, тоді:

- **Підзадача 0 [ 0 балів]**: Приклад.
- **Підзадача 1 [15 балів]**: Функція `change` ніколи не викликається;  $l = 0, r = N - 1$  у кожному виклику `perturb`.
- **Підзадача 2 [16 балів]**:  $A_i \leq 10$  для всіх  $0 \leq i < N$ , а  $x \leq 10$  у всіх викликах `change`.
- **Підзадача 3 [13 балів]**: Виклики `change` не зменшують значення ( $x \geq A_p$ ),  $Q_1 \leq 1000$ , та  $l = 0, r = N - 1$  у кожному виклику `perturb`.
- **Підзадача 4 [22 балів]**: Функція `change` ніколи не викликається.
- **Підзадача 5 [14 балів]**: Виклики `change` не зменшують значення ( $x \geq A_p$ ),  $Q_1 \leq 1000$ .
- **Підзадача 6 [20 балів]**: Без додаткових обмежень.

## Приклади вводу/виводу

stdin	stdout
10 28	1
5 1 7 8 3 2 5 6 9 4	3
1 1 1	1
1 0 1	7
2 0 1	8
2 2 6	1
1 6 5	8
2 2 9	3
2 2 5	6
2 4 5	4
1 4 5	9
2 3 8	
1 8 4	
3 0	
1 6 3	
1 4 1	
2 5 7	
1 0 3	
2 4 5	
1 6 3	
3 0	
3 1	
3 2	
3 3	
3 4	
3 5	
3 6	
3 7	
3 8	
3 9	

## Пояснення

Ми починаємо з масиву  $A = [5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ .

- Подія 1: Валентин змінює  $A_1$  на 1 (він уже був 1): новий масив  $[5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ .
- Подія 2: Валентин змінює  $A_0$  на 1: новий масив  $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ .
- Подія 3: Валентин збурює  $[0, 1]$ : новий масив  $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ .
- Подія 4: Валентин збурює  $[2, 6]$ : новий масив  $[1, 1, 7, 8, 8, 3, 5, 6, 9, 4]$ .

Починаючи з події 19, Валентин лише запитує значення в масиві, не виконуючи змін чи збурень. На цей момент масив дорівнює  $[3, 1, 7, 8, 1, 8, 3, 6, 4, 9]$ .