

## Wild operations (wild)

Дамјан сака да ја тестира способноста на Владимир да се справува со луди операции со низи, па затоа му дал низа  $A_0, \dots, A_{N-1}$  со должина  $N$ .

Сега Дамјан ќе побара од Владимир да изврши некои операции врз низата, каде што секоја операција може да биде:

- *промена* на вредноста на  $A_p$  во  $x$ , за некој цел број  $x$  и валиден индекс  $p$ .
- *пертурбација* на опсегот  $[l, r]$ , т.е. да се постави  $A_p = \max(A_p, A_{p-1})$  **истовремено** за сите  $l < p \leq r$ .

Во кој било момент, Дамјан може да побара од Владимир да му ја каже вредноста на  $A_p$  за некој валиден индекс  $p$ .

Владимир е многу зафатен, па затоа одлучил да побара твоја помош за да одговори на прашањата на Дамјан.

## Имплементација

Треба да предадеш една датотека со екстензија `.cpp`.



Меѓу додатоците (анг. attachments) за оваа задача ќе најдеш темплејт `wild.cpp` со пример за имплементација.

Треба да ги имплементираш следниве функции:

C++

```
void init(int N, vector<int> A);
```

- Оваа функција се повикува еднаш, на почетокот на извршувањето на твојата програма.
- Целиот број  $N$  е должината на низата.
- Низата  $A$ , индексирана од 0 до  $N - 1$ , е почетната низа што ја избрал Дамјан.

C++

```
void change(int p, int x);
```

- Оваа функција се повикува повеќе пати во текот на извршувањето на твојата програма, кога Дамјан врши промена.
- Целиот број  $p$  е индексот на променетата вредност во низата.
- Целиот број  $x$  е новата вредност што треба да се додели.

C++

```
void perturb(int l, int r);
```

- Оваа функција се повикува повеќе пати во текот на извршувањето на твојата програма, кога Дамјан прави пертурбација на опсег.
- Целиот број  $l$  е левиот крај на опсегот на кој Дамјан прави пертурбација.
- Целиот број  $r$  е десниот крај на опсегот на кој Дамјан прави пертурбација.

C++

```
int calc(int p);
```

- Оваа функција се повикува повеќе пати во текот на извршувањето на твојата програма, кога Дамјан прашува за вредноста на елемент од низата.
- Целиот број  $p$  е индексот на елементот што го бара Дамјан.
- Функцијата треба да ја врати вредноста на  $A_p$  по примената на сите претходни операции.

## Пример-оценувач

Поедноставена верзија на оценувачот што се користи при оценувањето е достапна во фолдерот за оваа задача. Можеш да ја користиш за локално тестирање на твоите решенија. Пример-оценувачот чита влезни податоци од `stdin`, ги повикува функциите што треба да ги имплементираш, и запишува во `stdout` во следниов формат.

Нека  $Q$  е вкупниот број на промени, пертурбации и прашања направени од Дамјан. Тогаш, влезната датотека се состои од  $2 + Q$  линии, кои содржат:

- Линија 1: целите броеви  $N, Q$ .
- Линија 2:  $N$  цели броеви  $A_0, \dots, A_{N-1}$ , почетните вредности на низата.
- Линија  $3 + i$  ( $0 \leq i < Q$ ): 2 или 3 цели броеви, во еден од следниве формати:
  - $1\ p\ x$ : значи дека Дамјан го менува  $A_p$  во  $x$ .
  - $2\ l\ r$ : значи дека Дамјан прави пертурбација на опсегот  $[l, r]$ ;
  - $3\ p$ : значи дека Дамјан прашува за вредноста на  $A_p$ .

Излезната датотека се состои од  $Q_3$  линии (каде  $Q_3$  е бројот на повици на `calc`) кои ги содржат вредностите вратени од функцијата `calc`.

## Ограничувања

- $1 \leq N \leq 400\,000$ .
- $0 \leq Q \leq 400\,000$ .
- $1 \leq A_i \leq 10^9$  за сите  $0 \leq i < N$ .
- $0 \leq p < N$  во секој повик на `change` и `calc`.
- $0 \leq l < r \leq N - 1$  во секој повик на `perturb`.
- $1 \leq x \leq 10^9$  во секој повик на `change`.

## Бодување

Твојата програма ќе биде тестирана на неколку тест случаи групирани во подзадачи. За да ги освоиш поените за една подзадача, мораш точно да ги решиш сите тест случаи во неа.

Нека  $Q_1$  е бројот на повици на функцијата `change` во еден тест случај, тогаш:

- **Потзадача 0 [ 0 поени]**: Пример.
- **Потзадача 1 [15 поени]**: Функцијата `change` никогаш не се повикува;  $l = 0, r = N - 1$  во секој повик на `perturb`.
- **Потзадача 2 [16 поени]**:  $A_i \leq 10$  за сите  $0 \leq i < N$  и  $x \leq 10$  за сите повици на `change`.
- **Потзадача 3 [13 поени]**: Повиците на функцијата `change` не ги намалуваат вредностите ( $x \geq A_p$ ),  $Q_1 \leq 1000$  и  $l = 0, r = N - 1$  во секој повик на `perturb`.
- **Потзадача 4 [22 поени]**: Функцијата `change` никогаш не се повикува.
- **Потзадача 5 [14 поени]**: Повиците на функцијата `change` не ги намалуваат вредностите ( $x \geq A_p$ ),  $Q_1 \leq 1000$ .
- **Потзадача 6 [20 поени]**: Нема дополнителни ограничувања.

## Примери за влез/излез

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	

## Објаснување

Почнуваме со низата  $A = [5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ .

- Настан 1: Дамјан го менува  $A_1$  во 1 (веќе беше 1): новата низа е  $[5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ .
- Настан 2: Дамјан го менува  $A_0$  во 1: новата низа е  $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ .
- Настан 3: Дамјан прави пертурбација на  $[0, 1]$ : новата низа е  $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ .
- Настан 4: Дамјан прави пертурбација на  $[2, 6]$ : новата низа е  $[1, 1, 7, 8, 8, 3, 5, 6, 9, 4]$ .

Од настан 19 натаму, Дамјан само прашува за вредности во низата без да врши промени или пертурбации. Во овој момент, низата е  $[3, 1, 7, 8, 1, 8, 3, 6, 4, 9]$ .