Wild operations (wild)

Jovan želi da testira Lukinu sposobnost snalaženja sa čudnim operacijama nad nizovima, pa mu je dao niz $A_0, ..., A_{N-1}$ dužine N.

Sada će Jovan tražiti od Luke da izvrši neke operacije nad nizom, pri čemu svaka operacija može biti:

- *promjena* vrijednosti A_p u x, za neki cijeli broj x i validan indeks p.
- *perturbacija* opsega $[l, r]$, tj. postavljanje $A_p = \max(A_p, A_{p-1})$ **istovremeno** za sve $l < p \leq r$.

U bilo kojem trenutku, Jovan može pitati Luku da mu kaže vrijednost A_p za neki validan indeks p.

Kako je Luka veoma zauzet, odlučio je da vas zamoli za pomoć u odgovaranju na Jovanova pitanja.

Implementacija

Morate predati jednu datoteku sa ekstenzijom `.cpp`.

⇦ Među prilozima za ovaj zadatak nalazi se šablon `wild.cpp` sa primjerom implementacije.

Potrebno je implementirati sljedeće funkcije:

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

- Ova funkcija se poziva jednom, na samom početku izvršavanja vašeg programa.
- Cijeli broj N je dužina niza.
- Niz A, indeksiran od 0 do $N-1$, je početni niz koji je Jovan odabrao.

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

- Ova funkcija se poziva više puta tokom izvršavanja vašeg programa, kada Jovan izvrši promjenu.
- Cijeli broj p je indeks promijenjene vrijednosti u nizu.
- Cijeli broj x je nova vrijednost koja se dodjeljuje.

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

- Ova funkcija se poziva više puta tokom izvršavanja vašeg programa, kada Jovan izvrši perturbaciju opsega.
- Cijeli broj l je lijevi kraj opsega koji Jovan perturbira.
- Cijeli broj r je desni kraj opsega koji Jovan perturbira.

C++
```
int calc(int p);
```

- Ova funkcija se poziva više puta tokom izvršavanja vašeg programa, kada Jovan pita za vrijednost elementa niza.
- Cijeli broj p je indeks elementa za koji Jovan pita.
- Funkcija treba da vrati vrijednost A_p nakon primjene svih prethodnih operacija.

Primjer gradera

Pojednostavljena verzija gradera koji se koristi tokom ocjenjivanja dostupna je u direktoriju vezanom za ovaj zadatak. Možete je koristiti da testirate svoja rješenja lokalno. Primjer gradera čita ulazne podatke iz `stdin`, poziva funkcije koje trebate implementirati i ispisuje u `stdout` u sljedećem formatu.

Neka je Q ukupan broj promjena, perturbacija i pitanja koje je postavio Jovan.
Ulazna datoteka se onda sastoji od $2 + Q$ linija, koje sadrže:

- Linija 1: cijeli brojevi N, Q.
- Linija 2: N cijelih brojeva $A_0, ..., A_{N-1}$, početne vrijednosti niza.
- Linija $3 + i$ $(0 \leq i < Q)$: 2 ili 3 cijela broja, u jednom od sljedećih formata:
  - 1 p x: znači da Jovan mijenja A_p u x.
  - 2 l r: znači da Jovan perturbira opseg $[l, r]$;
  - 3 p: znači da Jovan pita za vrijednost A_p.

Izlazna datoteka se sastoji od Q_3 linija (gdje je Q_3 broj poziva funkcije `calc`) koje sadrže vrijednosti vraćene od strane funkcije `calc`.

Ograničenja

- $1 \leq N \leq 400\,000$.
- $0 \leq Q \leq 400\,000$.
- $1 \leq A_i \leq 10^9$ za sve $0 \leq i < N$.
- $0 \leq p < N$ u svakom pozivu funkcija `change` i `calc`.
- $0 \leq l < r \leq N - 1$ u svakom pozivu funkcije `perturb`.
- $1 \leq x \leq 10^9$ u svakom pozivu funkcije `change`.

Bodovanje

Vaš program će biti testiran na nekoliko testnih primjera grupisanih u podzadatke. Da biste dobili bodove za podzadatak, morate tačno riješiti sve testne primjere u njemu.

Neka je Q_1 broj poziva funkcije `change` u jednom testnom primjeru, tada:

- **Podzadatak 0 [0 bodova]**: Testni primjer.
- **Podzadatak 1 [15 bodova]**: Funkcija `change` se nikada ne poziva; $l = 0$, $r = N - 1$ u svakom pozivu funkcije `perturb`.
- **Podzadatak 2 [16 bodova]**: $A_i \leq 10$ za sve $0 \leq i < N$ i $x \leq 10$ za sve pozive funkcije `change`.
- **Podzadatak 3 [13 bodova]**: Pozivi funkcije `change` ne smanjuju vrijednosti $(x \geq A_p)$, $Q_1 \leq 1000$ i $l = 0$, $r = N - 1$ u svakom pozivu funkcije `perturb`.
- **Podzadatak 4 [22 bodova]**: Funkcija `change` se nikada ne poziva.
- **Podzadatak 5 [14 bodova]**: Pozivi funkcije `change` ne smanjuju vrijednosti $(x \geq A_p)$, $Q_1 \leq 1000$.
- **Podzadatak 6 [20 bodova]**: Nema dodatnih ograničenja.

Primjeri ulaza/izlaza

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

Objašnjenje

Počinjemo sa nizom $A = [5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.

- Događaj 1: Jovan mijenja A_1 u 1 (već je bio 1): novi niz je $[5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Događaj 2: Jovan mijenja A_0 u 1: novi niz je $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Događaj 3: Jovan perturbira $[0, 1]$: novi niz je $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Događaj 4: Jovan perturbira $[2, 6]$: novi niz je $[1, 1, 7, 8, 8, 3, 5, 6, 9, 4]$.

Od događaja 19 pa nadalje, Jovan pita samo za vrijednosti u nizu bez vršenja promjena ili perturbacija. U ovom trenutku, niz je $[3, 1, 7, 8, 1, 8, 3, 6, 4, 9]$.