

Zadanie: STO

Stos



ONTAK 2021, dzień drugi. Dostępna pamięć: 512 MB. Limit czasu: 3 s.

29.06.2021

Bajtokomputer to najnowszy bajtowski wynalazek. Jest to bardzo wydajna maszyna, ale posiada przy tym pewien haczyk. Otóż bajtokomputer od innych komputerów różni się tym, że jest jednowątkowy, a wszystkie zapytania przez niego wykonywane najpierw muszą zostać wrzucone na *stos zapytań*. Jak wygląda to w praktyce? Bajtokomputer w każdym momencie może wykonywać jedną z dwóch operacji:

- przyjąć zapytanie x – czyli wrzucić zapytanie o numerze x na szczyt stosu zapytań;
- odpowiedzieć na zapytanie – a więc wziąć pewne zapytanie ze szczytu stosu zapytań, odpowiedzieć na nie, a następnie wyrzucić najwyżej położony element ze stosu zapytań.

Bajtazar zakupił niedawno jeden z bajtokomputerów z początkowo pustym stosem zapytań i postanowił przetestować jego działanie. Przygotował więc ciąg operacji, które oznaczył jako c_1, c_2, \dots, c_N . Dla i -tej operacji równość $c_i = -1$ oznacza odpowiedź na zapytanie obecnie znajdujące się na szczycie stosu zapytań, zaś $c_i \geq 0$ oznacza wrzucenie zapytania o numerze c_i na stos zapytań. Bajtazar chce być bardzo dokładny w swym teście, dlatego przygotował aż $10^6 + 1$ potencjalnych zapytań dla bajtokomputera, przez co elementy ciągu c_i są z przedziału od -1 do 10^6 włącznie.

Kolejnym krokiem Bajtazara było przygotowanie dwóch ciągów a_1, a_2, \dots, a_M oraz b_1, b_2, \dots, b_M , gdzie $1 \leq a_j \leq b_j \leq N$ dla każdego $1 \leq j \leq M$. Bajtazar chce przeprowadzić test, wykonując operacje z ciągu c_j na kolejnych przedziałach $[a_i, b_i]$. Oznacza to, że bajtokomputer będzie musiał najpierw wykonać kolejne operacje $c_{a_1}, c_{a_1+1}, c_{a_1+2}, \dots, c_{b_1}$, następnie kolejne operacje $c_{a_2}, c_{a_2+1}, c_{a_2+2}, \dots, c_{b_2}$, zaś ostatnią operacją wykonaną przez bajtokomputer będzie c_{b_M} . Bajtazar przygotował test bardzo dokładnie, dlatego masz gwarancję, że w żadnym momencie bajtokomputer nie będzie odpowiadał na zapytanie z pustym stosem zapytań.

Bajtazar nie wykonał jeszcze testu, jednak chciałby już teraz wiedzieć, jaki powinien być wynikowy stan stosu zapytań po przeprowadzeniu testu. Niestety zbyt szybko pozbył się swojego zwykłego komputera i nie jest w stanie przeprowadzić symulacji. Dlatego zwrócił się o pomoc do Ciebie, przekazując Tobie ciągi c_i, a_j, b_j , a także pewien rosnący ciąg liczb naturalnych p_1, p_2, \dots, p_K . Dla każdego $1 \leq k \leq K$ Bajtazar po przeprowadzeniu symulacji chciałby poznać p_k -ty element znajdujący się na stosie zapytań, licząc od najniższej położonego elementu. Czy pomożesz Bajtazarowi?

Wejście

Pierwszy wiersz wejścia zawiera trzy liczby całkowite N, M, K oznaczające odpowiednio: długość ciągu c_i , długość obydwu ciągów a_j, b_j , oraz długość ciągu p_k .

W kolejnych N wierszach wejścia znajdują się kolejne elementy ciągu c_1, c_2, \dots, c_N . W następnych M wierszach znajdują się kolejne pary liczb a_j, b_j oddzielonych pojedynczym odstępem.

Ostatnie K wierszy wejścia zawiera kolejne elementy *rosnącego* ciągu p_1, p_2, \dots, p_K . Masz gwarancję, że po przeprowadzeniu symulacji na stosie zapytań będzie znajdował się element o numerze p_K .

Wyjście

Na wyjście wypisz dokładnie K wierszy, w k -tym z nich pojedynczą liczbę całkowitą oznaczającą numer p_k -tego zapytania znajdującego się na wynikowym stosie zapytań.

Ograniczenia

Dla wszystkich podzadań $1 \leq N, M, K \leq 200\,000$ oraz $1 \leq p_k \leq 2 \cdot 10^{10}$ dla $1 \leq k \leq K$. W poniższej tabeli podane są dodatkowe ograniczenia dla podzadań:

Podzadanie	Ograniczenia	Punkty
1	$N, M \leq 2000$ oraz $p_k \leq 2 \cdot 10^6$	15
2	wszystkie elementy ciągu c_i są nieujemne	15
3	$N \leq 10\,000$, N jest parzyste oraz dla każdego j zachodzi $b_j - a_j = N/2$	30
4	brak dodatkowych ograniczeń	40

Przykład

Dla danych wejściowych:

4 3 2
-1
5
6
-1
2 4
1 3
1 4
1
2

poprawnym wynikiem jest:

5
5