

Pojedynek (A)

Limit pamięci: 256 MB

Limit czasu: 3.00 s

1. Pewnego jesiennego dnia Frymbulin miał ochotę na zupę grzybową. W związku z tym, że zupę grzybową gotuje się głównie z grzybów, wziął koszyczek i poszedł w głąb okolicznych lasów na grzyby.
2. Podczas zbierania grzybów w lesie, Frymbulin potknął się, upadł i czubkiem nosa wbił się w mech. Kiedy w końcu udało mu się wydostać nos z niewoli, zobaczył, że pod nim znajduje się ruda żelaza.

Frymbulin przypomniał sobie wtedy o swoim zamiłowaniu do rozważań ciekawych operacji na ciągach. Niestety, jest jeszcze częściowo oszołomiony po wyjściu z opresji, w której się znalazł, dlatego tym razem rozważane przez niego operacje są nieco bezsensowne.

Dany jest ciąg długości N , składający się z elementów ponumerowanych od 1 do N .

Na ciąg zostanie wykonanych Q operacji, z których każda będzie jednego z dwóch typów:

- 1 i r – operacja oznaczająca, że element na pozycji i zmienia swoją wartość na r .
- 2 l k – zapytanie o wynik poniższego procesu na przedziale naszego ciągu długości 2^k , zaczynającym się od pozycji l i kończącym się na pozycji $l + 2^k - 1$:
 - Proces będzie przebiegać w k krokach. Każdy krok zmniejsza długość ciągu o połowę. W danym kroku wartości są łączone w pary kolejnych (pierwsza i druga, trzecia i czwarta itd.), a następnie każda para jest zastępowana jedną wartością równą wartości bezwzględnej różnicy między nimi. Dokładniej, jeśli w jednej parze będą wartości A i B , to nowa wartość wyniesie $|A - B|$. Zauważ, że po k takich krokach pozostanie dokładnie **jeden** element. Wynikiem jest jego wartość.

Operacja typu 2 **nie zmienia ciągu** – wszystkie kroki wykonywane są tylko abstrakcyjnie. Należy wypisać wynik dla każdej operacji typu 2.

Wejście

W pierwszym wierszu znajdują się dwie liczby całkowite N i Q .

W następnym wierszu znajduje się N liczb całkowitych p_i oznaczających elementy ciągu.

W kolejnych Q wierszach znajdują się opisy zapytań zgodnie z treścią zadania.

Wyjście

Dla każdego zapytania typu 2 należy wypisać końcową wartość po wszystkich k krokach.

Ograniczenia

$2 \leq N \leq 200\,000$, $1 \leq Q \leq 200\,000$, $0 \leq p_i \leq 10^9$.

Dla zapytań typu 1 zachodzi $1 \leq i \leq N$, $0 \leq r \leq 10^9$.

Dla zapytań typu 2 zachodzi $1 \leq l \leq N$, $1 \leq l + 2^k - 1 \leq N$.

Podzadania

Podzadanie	Ograniczenia	Punkty
1	$N, Q \leq 1000$	11
2	Dla wszystkich zapytań typu 2 zachodzi $2^k = N$	13
3	$p_i \leq 1$ oraz dla wszystkich zapytań typu 1 zachodzi $r \leq 1$	16
4	Brak zapytań typu 1	17
5	Brak dodatkowych ograniczeń	43

Testy przykładowe

Wejście

5 3
4 8 2 0 7
2 1 2
1 1 9
2 2 1

Wyjście

2
6

Wejście

8 6
1 2 3 4 5 6 7 8
2 1 3
1 4 1
1 7 3
2 1 3
1 2 100
2 2 2

Wyjście

0
3
93

Wejście

9 5
1 0 2 0 4 1 3 2 8
2 2 3
2 1 3
1 5 1
1 6 4
2 4 2

Wyjście

2
1
0

Wyjaśnienie

W pierwszym zapytaniu pierwszego testu wartości będą się zmieniać następująco: $(4, 8, 2, 0) \rightarrow (4, 2) \rightarrow (2)$

W trzecim zapytaniu pierwszego testu wartości będą się zmieniać następująco:

$(8, 2) \rightarrow (6)$

Sortowanie (B)

Limit pamięci: 256 MB

Limit czasu: 4.00 s

To zadanie jest interaktywne. Po wypisaniu każdego wiersza należy opróżnić bufor. W C++ możesz użyć `cout << flush`, w Pythonie – `sys.stdout.flush()`. Należy ściśle trzymać się protokołu opisanego w sekcji *Interakcja* – niewczytanie wiadomości wysłanej przez interaktor skutkuje werdyktem *wrong answer* lub innym.

-
3. Frymbulin włożył do koszyczka z grzybami kilka kawałków rudy żelaza. Kiedy wrócił do domu, zaczął zastanawiać się, do czego mogłoby się przydać jego znalezisko. A dlatego, że to był skrzatek bardzo sprytny, zaraz wpadł mu do głowy pewien pomysł.
 4. Zbudował prosty piec na węgiel drzewny i wytopił w nim żelazo. Potem wykuł nóż, pokroił grzyby i ugotował z nich pyszną zupę. Mniam mniam.

Oprócz zupy grzybowej, Frymbulin bardzo lubi sortowanie przez scalanie. W tym celu potrzeba jednak stworzyć procedurę merge. Jako że komputer zbudowany z żelaza ma trochę inne operacje niż zwykły, nie jest to trywialne zadanie. Musisz mu w tym pomóc.

Dane są liczby N, M oraz pewien ukryty ciąg długości $N + M$, składający się z parami różnych liczb. Dodatkowo, wiadomo o nim, że powstał on poprzez połączenie (jeden po drugim) dwóch rosnących ciągów – pierwszego o długości N i drugiego o długości M .

Należy posortować ten ciąg, używając następujących operacji:

- $? i j$ oznacza porównanie dwóch różnych elementów na pozycjach i oraz j . W odpowiedzi dostajemy znak $<$ lub $>$, w zależności od tego, który z elementów jest większy
- $\wedge i j$ oznacza odwrócenie fragmentu między pozycjami i oraz j włącznie

Istnieją limity na łączną liczbę operacji oraz na sumę długości odwracanych fragmentów. **Zależą one od podzadania**, podobnie jak limity na N i M .

Interakcja

Interaktor nie jest adaptacyjny – ciągi są ustalone na początku i nie zmieniają się w zależności od zadanych zapytań.

Na początku należy wczytać liczby całkowite N, M, L_{oper}, L_{sum} . Wartości L_{oper} i L_{sum} oznaczają limit na liczbę operacji i sumę długości odwracanych fragmentów.

Następnie należy wykonać ciąg operacji, każdą poprzez wypisanie odpowiedniego wiersza, a następnie **opróżnienie bufora**. Każda operacja zaczyna się od znaku $?$ lub $\wedge$.

Jeśli operacja to porównanie, po $?$ powinny nastąpić dwa indeksy $1 \leq i \neq j \leq N + M$ porównywanych elementów. Następnie powinien być wczytany znak $>$ lub $<$ – odpowiedź na zapytanie ($>$ jeśli element na pozycji i jest większy, w przeciwnym przypadku $<$).

Jeśli operacja to odwrócenie, po znaku $\wedge$ powinny nastąpić dwa indeksy $1 \leq i \leq j \leq N + M$ – krańce odwracanego przedziału.

Po wykonaniu wszystkich operacji należy wypisać znak $!$, opróżnić bufor i zakończyć działanie programu. Nie należy już nic więcej wypisywać. Jeśli ciąg został posortowany i zmieściliśmy się w limitach, test zostanie zaakceptowany.

Ograniczenia

We wszystkich podzadaniach zachodzi $2 \leq N, M \leq 50\,000$. Pierwszych N elementów ciągu tworzy ciąg rosnący, podobnie jak ostatnich M elementów.

Podzadania

Poniższa tabelka zawiera ograniczenia na N, M, L_{oper}, L_{sum} w każdym z podzadań.

	N	M	L_{oper}	L_{sum}	Punkty
1	50	50	100 000	3 000 000	18
2	500	500	100 000	3 000 000	11
3	500	5 000	100 000	3 000 000	8
4	500	50 000	100 000	3 000 000	24
5	5 000	5 000	50 000	3 000 000	13
6	5 000	5 000	100 000	200 000	14
7	50 000	50 000	300 000	3 000 000	12

Przykładowa interakcja

Wejście	Wyjście	Wyjaśnienie
2 3 100000 3000000		Ciąg to (1, 5, 2, 3, 4)
>	? 2 5	5 > 4
	^ 2 5	Odwracamy fragment (5, 2, 3, 4), ciąg teraz to (1, 4, 3, 2, 5)
>	? 3 1	3 > 1
<	? 1 4	1 < 2
	^ 2 4	Odwracamy fragment (4, 3, 2), ciąg teraz to (1, 2, 3, 4, 5)
	!	Ciąg jest posortowany poprawnie

Odpowiada ona **pierwszemu testowi przykładowemu**. Liczba wykonanych operacji to 5 (3 porównania i 2 odwrócenia), natomiast suma długości to $4 + 3 = 7$.

Narzędzia do testowania lokalnego i testy przykładowe

W zadaniu są 2 testy przykładowe. Zostanie również udostępniony program **interaktor.cpp** oraz przykładowe błędne rozwiązanie o poprawnym schemacie komunikacji **abc.cpp**. Aby uruchomić rozwiązanie na teście, należy skompilować rozwiązanie i interaktor, a następnie użyć komendy:

```
[plik wykonywalny z interaktor.cpp] [test] [plik wykonywalny z rozwiązaniem]
```

Na przykład: `./interaktor 0a.in ./abc`