Sortowanie

XIX OIJ, zawody trzeciego stopnia – dzień pierwszy
12 kwietnia 2025

Kod zadania: **sor**
Limit czasu: **5 s (C++) / 15 s (Python)**
Limit pamięci: **256 MiB**

Olimpiada Informatyczna Juniorów

Bajtek pracuje w księgowości. Zbiera dokumentację, oblicza sumy, uzupełnia tabelki, studiuje ciągle zmieniające się przepisy podatkowe, a czasami prezentuje dane klientom w wygodnej dla nich formie.

Dzisiaj Bajtek musi posortować ciąg wyników finansowych pewnej firmy za kolejne miesiące. W dużym uproszczeniu chodzi o uporządkowanie ciągu liczb A od najmniejszych do największych. Z tymi liczbami oczywiście związane są rozliczenia finansowe za poszczególne miesiące, ale Bajtek nie chce Ci tym zawracać głowy. Jeśli w ciągu Bajtka znajdują się elementy równe, w uporządkowanym ciągu mogą się znaleźć w dowolnej kolejności względem siebie.

Niestety, oprogramowanie księgowe, którego używa Bajtek, udostępnia jedynie sortowanie za pomocą przenoszenia na początek ciągu elementów znajdujących się na zaznaczonych pozycjach. Pozycje elementów w ciągu są oznaczone kolejnymi liczbami naturalnymi, zaczynając od 1.

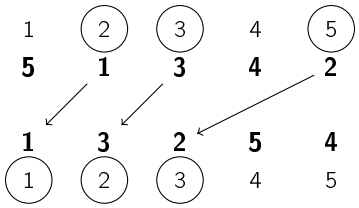
Dokładniej, oprogramowanie pozwala wykonywać operacje trzech typów:

- `zaznacz`(p), która zaznacza pozycję p,
- `odznacz`(p), która odznacza pozycję p,
- `przesun`, która przesuwa liczby z zaznaczonych pozycji na początek ciągu.

Bajtek nie może zaznaczyć pozycji już zaznaczonej, ani odznaczyć pozycji, która nie jest zaznaczona. Po wykonaniu operacji `przesun`, liczby z zaznaczonych pozycji nie zmieniają kolejności między sobą. Liczby z niewybranych pozycji też nie zmieniają kolejności między sobą. Jeżeli przed wykonaniem tej operacji zaznaczonych było K pozycji, to po jej wykonaniu, zbiór zaznaczonych pozycji zmieni się na $\{1, 2, \ldots, K\}$.

Przykładowo: jeżeli ciąg A to $(5, 1, 3, 4, 2)$, a Bajtek zaznaczy elementy na pozycjach drugiej, trzeciej i piątej, a następnie wykona operację `przesun` to uzyska ciąg $(1, 3, 2, 5, 4)$ Zaznaczone wtedy będą pozycje $1, 2, 3$.

Na poniższym rysunku zobrazowano przebieg operacji `przesun` w tym przykładzie.



Pomóż Bajtkowi uporządkować jego ciąg niemalejąco (tzn. od najmniejszych liczb do największych). Im mniej operacji wykonasz, tym więcej punktów możesz uzyskać za to zadanie.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N $(1 \leq N \leq 100\,000)$ oznaczająca liczbę elementów ciągu A. W drugim (ostatnim) wierszu wejścia znajduje się ciąg N liczb naturalnych A_i $(1 \leq A_i \leq 10^9)$, pooddzielanych pojedynczymi odstępami.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

Wyjście

Na pierwszy wiersz wyjścia należy wypisać jedną liczbę naturalną R, oznaczającą liczbę wykonanych operacji. W kolejnych R wierszach powinien się znaleźć opis kolejno wykonanych operacji. Każdy z wypisanych wierszy opisu powinien być w jednym z następujących formatów:

- `zaznacz` p, gdzie $1 \leq p \leq N$ oznacza pozycję do zaznaczenia;
- `odznacz` p, gdzie $1 \leq p \leq N$ oznacza pozycję do odznaczenia;
- `przesun`.

Jeśli istnieje wiele ciągów operacji prowadzących do posortowania ciągu, Twój program może wypisać dowolną. Liczba uzyskanych punktów będzie zależała od liczby wypisanych operacji, zgodnie z informacją poniżej, w sekcji *Ocenianie*. Zauważ, że Twój program **nie musi** wypisywać ciągu z najmniejszą liczbą operacji.

Ocenianie

Na każdym teście, w którym Twój program zakończy się poprawnie, zmieści z limitach czasu i pamięci, a także poprawnie posortuje ciąg A, otrzymasz punkty zgodnie z poniższą tabelą:

R	punkty
co najwyżej $200\,000$	100% punktów za test
co najwyżej $300\,000$	80% punktów za test
co najwyżej $400\,000$	60% punktów za test
co najwyżej $2\,000\,000$	40% punktów za test
więcej niż $2\,000\,000$	0% punktów za test i werdykt *Zła odpowiedź*

Możesz rozwiązać zadanie w kilku prostszych wariantach – niektóre grupy testów spełniają pewne dodatkowe ograniczenia. Poniższa tabela pokazuje, ile punktów otrzyma Twój program, jeśli przejdzie testy z takim ograniczeniem.

Dodatkowe ograniczenia	**Liczba punktów**
$N \leq 8$	20
$N \leq 1\,000$	35
ciąg A jest nierosnący, tzn. $A_1 \geq A_2 \geq \ldots \geq A_N$	15
każda z liczb od 1 do N występuje w ciągu A dokładnie raz	40

Przykłady

Wejście dla testu `sor0a`:

```
5
5 1 3 4 2
```

Wyjście dla testu `sor0a`:

```
9
zaznacz 2
zaznacz 5
zaznacz 3
przesun
odznacz 2
przesun
zaznacz 3
zaznacz 5
przesun
```

Wyjaśnienie do przykładu: Pierwsza operacja `przesun` jest taka sama, jak opisana powyżej. Po drugiej operacji `przesun` ciąg wygląda tak: $(1, 2, 3, 5, 4)$. Po trzeciej operacji ciąg jest uporządkowany: $(1, 2, 3, 4, 5)$. Zauważ, że Bajtek nie musi odznaczać zaznaczonych elementów.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

Wejście dla testu `sor0b`:

```
5
1 4 4 9 16
```

Wyjście dla testu `sor0b`:

```
0
```

Wyjaśnienie do przykładu: Ciąg z wejścia jest już uporządkowany. Nie trzeba zatem wykonywać żadnej operacji. Oczywiście nic nie stoi na przeszkodzie, żeby wykonać jakieś operacje. Po wykonaniu ich wszystkich ciąg musi być uporządkowany.

Pozostałe testy przykładowe

- test `sor0c`: $N = 1\,000$, elementy ciągu A są równe, w kolejności, $1000, 999, 998, 997, \ldots, 2, 1$,
- test `sor0d`: $N = 100\,000$, elementy ciągu A są równe, w kolejności, $1, 2, 1, 2, 1, 2, \ldots$.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".