

# E. Liczby

---

Dostępna pamięć: 8 MB

Zaprojektuj i zaimplementuj strukturę, która umożliwi przechowywanie zbioru liczb całkowitych  $\mathcal{P}$  i wykonywanie na nim opisanych poniżej operacji.

1. `INSERT(x)`. Dodaje liczbę całkowitą  $x$  do zbioru  $\mathcal{P}$ . Jeśli  $x$  już należy do  $\mathcal{P}$ , nic się nie dzieje.
2. `DELETE(x)`. Usuwa liczbę całkowitą  $x$  ze zbioru  $\mathcal{P}$ .
3. `UPPER(x)`. Zwraca liczbę  $y \in \mathcal{P}$ , taką że  $y \geq x$  i  $y$  jest najmniejszą liczbą o takiej własności.
4. `LOWER(x)`. Zwraca liczbę  $y \in \mathcal{P}$ , taką że  $y \leq x$  i  $y$  jest największą liczbą o takiej własności.

**Uwaga:** W tym zadaniu zabronione jest używanie tych konstrukcji STL-a, których nazwy zawierają: `set`, `map` lub `hash`. Niedozwolone jest wykorzystanie gotowych odpowiedników tych konstrukcji w innych językach programowania.

## Specyfikacja danych wejściowych

W pierwszym wierszu danych wejściowych znajduje się liczba naturalna  $N \in [1, 10^6]$ , oznaczająca liczbę operacji na zbiorze  $\mathcal{P}$ . Początkowo zbiór  $\mathcal{P}$  jest pusty. W każdym z kolejnych  $N$  wierszy znajduje się opis jednej operacji wykonywanej na zbiorze  $\mathcal{P}$ . Każdy z wierszy składa się z dużej litery ze zbioru  $\{I, D, U, L\}$ , pojedynczego odstępu i liczby całkowitej  $x \in [-10^{18}, 10^{18}]$ . Podana litera jest pierwszą literą operacji zdefiniowanych powyżej. Operacje są tak dobrane, że po każdej z nich rozmiar zbioru wynosi co najwyżej 50 000.

## Specyfikacja danych wyjściowych

Twój program powinien wypisać jeden wiersz dla każdej operacji `DELETE`, `UPPER` lub `LOWER`. Zawartość tego wiersza powinna być następująca: dla operacji `DELETE(x)` należy wypisać słowo `BRAK`, jeśli  $x \notin \mathcal{P}$  i `OK` w przeciwnym przypadku; zaś dla operacji `UPPER` lub `LOWER` należy wypisać znaną liczbę, a jeśli taka nie istnieje — słowo `BRAK`.

## Przykład A

| Wejście: | Wyjście: |
|----------|----------|
| 4        | OK       |
| I 2      | BRAK     |
| D 2      | BRAK     |
| D 2      |          |
| D 2      |          |

## Przykład B

| Wejście: | Wyjście: |
|----------|----------|
| 7        | BRAK     |
| I -2     | -2       |
| I 0      | -2       |
| L -4     | 0        |
| L -2     | 0        |
| L -1     |          |
| L 0      |          |
| L 3      |          |

## Przykład C

Wejście:

6  
I 2  
I 2  
I 3  
D 2  
U 0  
L 0

Wyjście:

OK  
3  
BRAK