Limit pamięci: 64 MB, Limit czasu: x s (C++) / x s (Python)

Zarządca zadań

https://szkopul.edu.pl/problemset/problem/vGGTtjHNep6u-qwg1Nsndp_a/site

OLIMPIJSKIE KOŁO INFORMATYCZNE
https://kurs.oki.org.pl/

Jesteś inżynierem, który projektuje chmurę komputerową. Co to jest? W skrócie serwer na żądanie – na 5 minut, a może na 5 lat. To skomplikowany system. Ale Twój zespół odpowiada za jeden, ale za to kluczowy komponent nowej, ultralekkiej infrastruktury – zarządzanie zadaniami (ang. **Task Scheduler**) - przydzielającego moc obliczeniową serwerów.

Do serwera spływają miliony żądań od klientów z całego świata. Każde żądanie posiada swój priorytet. **Task Scheduler** musi pobrać do wykonania zadanie o najwyższym priorytecie. Jednak środowisko chmurowe jest bardzo dynamiczne – klienci w dowolnej chwili mogą zresetować połączenie lub ręcznie anulować oczekujące zadanie, zanim zostanie ono przetworzone przez procesor. W takim przypadku żądanie musi zostać natychmiast usunięte z kolejki oczekujących.

Napisz zarządcę zadań!

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita Q ($1 \le Q \le 5 \cdot 10^5$), oznaczająca liczbę zdarzeń w systemie. W kolejnych Q wierszach znajdują się opisy poszczególnych zdarzeń. Każde zdarzenie należy do jednego z trzech typów:

`D p` – (Dodaj) Do systemu trafia nowe zadanie o priorytecie *p*. Gwarantujemy, że w danym momencie nie ma w kolejce zadania o takim samym *p*.

`U p` – (Usuń) Klient anuluje zadanie o priorytecie *p*. Należy je z niej usunąć. Może być pomyłka i nie ma zadania o priorytecie *p* – program powinien pominąć to żądanie.

`W` – (Wykonaj) Procesor prosi o kolejne zadanie do wykonania. Należy znaleźć zadanie o największym priorytecie, usunąć je z kolejki i wypisać jego priorytet.

Początkowo kolejka zadań jest pusta. Zadanie o danym priorytecie usunięte przez zdarzenie `U` lub `W` może zostać ponownie dodane w zdarzeniu `D`.

Priorytety są liczbami całkowitymi: $1 \le p \le 10^9$.

Wyjście

Dla każdego zdarzenia typu **`W`** Twój program powinien wypisać w osobnym wierszu największy aktualnie priorytet **`p`**. Jeżeli w momencie zapytania kolejka jest pusta, program powinien wypisać słowo `PUSTO`.

Autor: Daniel Olkowski

Przykłady

Przykład 1

Wejście:

```
9
D 50
D 80
D 20
W
U 100
U 50
W
W
W
```

Wyjście:

```
80
20
PUSTO
PUSTO
```

Wyjaśnienie:

1. Dodajemy zadanie z priorytetem 50.
2. Dodajemy zadanie z priorytetem 80.
3. Dodajemy zadanie z priorytetem 20.
4. Wybieramy i usuwamy zadanie które ma najwyższy priorytet (80) – Wypisujemy 80
5. Klient anuluje zadanie z priorytetem 100. Nie ma takiego. Nie usuwamy go z kolejki.
6. Klient anuluje zadanie z priorytetem 50. Istnieje w kolejce. Usuwamy je z kolejki.
7. Wybieramy i usuwamy jedyne pozostałe zadanie z priorytetem 20 – Wypisujemy 20
8. Kolejka jest pusta, więc wypisujemy PUSTO.
9. Kolejka nadal jest pusta, więc ponownie wypisujemy PUSTO.

Przykład 2

Wejście:

```
9
D 90
D 70
W
U 70
D 10
D 70
W
W
W
```

Wyjście:

```
90
70
10
PUSTO
```

Wyjaśnienie:

Przy pierwszym zdarzeniu W wypisujemy 90 – największy priorytet.

Następnie anulujemy zadanie z priorytetem 70. Dodajemy zadania o priorytetach 10, 70. Kolejne zdarzenia W zwracają odpowiednio 70 i 10. Przy ostatnim zdarzeniu kolejka jest już pusta.

Ocenianie

Ograniczenia dodatkowe	Liczba punktów
Q = 1	10
Nie występują zdarzenia typu U	20
Q <= 100	30
Bez dodatkowych ograniczeń	40

Autor: Daniel Olkowski