

Opis zadania

Twoim zadaniem jest zaimplementowanie prostej gry logicznej zbliżonej do znanej gry 2048.

Uwagi wstępne

- Wszystkie indeksy pól na planszy są liczone od zera.
- Pisząc program możesz założyć, że wszystkie dane podawane na wejście programu są poprawne, tzn. indeksy wierszy i kolumn są z dozwolonego zakresu i nie przekraczają odpowiednich rozmiarów planszy.

Format danych wejściowych

Dane wejściowe składają się z dwóch sekcji: określenia początkowego rozmiaru tablicy oraz listy operacji do wykonania.

Rozmiar tablicy

W pierwszej linii wejścia znajduje się jedna liczba naturalna dodatnia M określająca liczbę wierszy i kolumn planszy. ($1 \leq M \leq 1024$). Przyjmujemy, że na początku gry plansza ma rozmiar $M \times M$. Na każdym polu planszy znajduje się liczba naturalna, nie przekraczająca 2^{31} . W dalszej części opisu zadania będziemy mówić, że pole w planszy jest puste, jeśli jest na nim liczba zero.

Operacje do wykonania

Kolejne linie wejścia będą zawierać operacje, które należy wykonać zgodnie z ich szczegółowym opisem znajdującym się poniżej. Każda z tych linii będzie się rozpoczynać od pojedynczej wielkiej litery ze zbioru $\{N, L, P, G, D, S, C, W, K\}$, po której znajduje się ciąg (może być pusty) liczb naturalnych.

Operacje modyfikujące planszę

Plansza gry może być przekształcana za pomocą następujących poleceń:

- $N \ x \ y \ w$, gdzie x, y to liczby naturalne mniejsze niż M , natomiast w jest pewną liczbą naturalną nie większą niż 2^{16} . Polecenie to wstawia wartość w na pozycję (x,y) w planszy.

•**L** - wykonuje operację przesunięcia w lewo wszystkich elementów na planszy. Dokładniej, na każdym wierszu planszy wykonujemy operacje, które są łącznie równoważne

- przesunięciu w lewo, czyli ułożeniu w kolejnych pozycjach tego wiersza począwszy od pozycji **0** wszystkich niezerowych elementów tego wiersza i uzupełnienie wolnych miejsc zerami
- kompresji wiersza: przeszukujemy wiersz **od początku** i po znalezieniu na pozycji **i** oraz **i+1** takich samych, niezerowych wartości, wstawiamy ich sumę na pozycję **i**, a na pozycję **i+1** wstawiamy zero.
- powtarzamy powyższe dwie czynności tak długo, aż w wierszu nie ma obok siebie dwóch takich samych i niezerowych wartości

Przykład: przesunięcie w lewo wiersza

211104042020

daje

418400000000

•**P** - wykonuje operację przesunięcia w prawo wszystkich elementów na planszy. Operacja jest analogiczna do **L**, przy czym przesuwanie jest w prawo, a w czasie kompresji przeszukujemy wiersz od końca i po napotkaniu dwóch takich samych liczb na pozycjach **i** oraz **i-1** wpisujemy ich sumę na pozycję **i** oraz zero na pozycję **i-1**. Następnie uzupełniamy wolne miejsca na początku wiersza zerami.

Przykład: przesunięcie w prawo wiersza

211104042020

daje

000000021248

•**G** - wykonuje operację przesunięcia w górę wszystkich elementów na planszy. Dokładniej, na każdej kolumnie planszy wykonujemy operacje analogiczne do tych opisanych dla wierszy dla operacji **L**.

•**D** - wykonuje operację przesunięcia w dół wszystkich elementów na planszy. Dokładniej, na każdej kolumnie planszy wykonujemy operacje analogiczne do tych opisanych dla wierszy dla operacji **P**.

Operacje wejścia-wyjścia

•**S** - należy wypisać na standardowe wyjście sumę wszystkich liczb znajdujących się aktualnie na planszy, a po niej znak końca linii.

•**W** - należy wypisać na standardowe wyjście aktualny stan planszy **p** w formacie

$$\begin{array}{cccc}
 p[0][0] & p[0][1] & \dots & p[0][M-1] \\
 \vdots & & & \\
 p[M-1][0] & p[M-1][1] & \dots & p[M-1][M-1]
 \end{array}$$

czyli elementy w wierszach oddzielamy pojedynczą spacją, a kolejne wiersze umieszczamy jeden pod drugim. Po ostatnim wierszu również wypisujemy znak końca linii.

•**C** **M** $p[0][0]$ $p[0][1]$... $p[0][M-1]$... $p[M-1][0]$ $p[M-1][1]$... $p[M-1][M-1]$ - operacja ta ma za zadanie wczytać ze standardowego wejścia nową planszę gry. Jej rozmiar jest podany jako parametr **M**. Po nim następuje **M**² liczb oddzielonych spacjami, które należy wczytać do kolejnych wierszy planszy. Na przykład, operacja

C 2 4 0 0 6

powinna zresetować planszę do rozmiaru 2 i uzupełnić ją danymi $p[0][0] = 4$, $p[0][1] = 0$, $p[1][0] = 0$, $p[1][1] = 6$

•**K** - po napotkaniu tego polecenia należy wydrukować na standardowe wyjście sumę wszystkich liczb znajdujących się aktualnie na planszy, a po niej znak końca linii oraz zakończyć działanie programu.

Sposób przesłania zadania.

Do systemu BaCa należy przesłać pojedynczy plik zawierający kod źródłowy (**nie skompilowany!**) rozwiązania. Proszę pamiętać o umieszczeniu w pierwszej linii pliku w komentarzu Państwa imienia i nazwiska.