

Zamiatanie

Limit pamięci: 256 MB

Bajtazar jest skromnym bibliotekarzem, a w czasie wolnym zajmuje się też programowaniem robotów. Ostatnio jego bibliotekę odwiedziła grupa ruchliwych dzieci i poprzerzastała książki na jego ulubionym regale. Regał może pomieścić maksymalnie n książek i jest aktualnie pełny.

Początkowo książka nr i stała na i -tym miejscu regału. Po odejściu dzieci na i -tym miejscu regału znajduje się książka nr a_i . Celem Bajtazara jest przywrócenie książek do początkowego ustawiania. Tak się składa, że ma on robota do porządkowania książek, który może wykonywać dwie operacje:

- U - operacja zamiatania "do góry". Polega ona na iterowaniu wskaźnika od 1 do $n-1$ i jeśli książka, na którą aktualnie wskazuje wskaźnik, ma większy numer od książki o jeden dalej, to zamienia te książki miejscami.
- D - operacja zamiatania "w dół". Polega na tym samym co operacja U, ale wskaźnik iteruje od $n-1$ do 1, czyli w przeciwną stronę.

Programem tego robota nazywamy ciąg złożony ze znaków U i D, który oznacza kolejne operacje. Uruchomienie takiego programu polega na wykonaniu po kolei operacji w nim zapisanych, od lewej do prawej.

Bajtazar napisał już część programu, tworząc ciąg złożony ze znaków U, D i ? o długości k . Teraz chciałby wiedzieć, jak szybko może przywrócić oryginalne ustawienie książek, czyli ile razy musi uruchomić dany program, aby książki były poprawnie ustawione. Dla danego programu liczbę potrzebnych uruchomień będziemy dalej nazywać wynikiem programu.

Aby pozyskać jakieś informacje statystyczne, zastanawia się, ile jest możliwych dokończeń programu, czyli zastąpień znaków ? znakami U i D, dających konkretne wyniki (gdy nie ma znaków ? jest 1 dokończenie, czyli brak zmian). Dla każdego potencjalnego wyniku od 0 do $n-1$ chciałby się dowiedzieć, ile jest możliwych dokończeń dających właśnie ten wynik. Jako że te liczby mogą być bardzo duże, wystarczy, że wypiszesz ich resztę z dzielenia przez 998244353.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite n oraz k ($1 \leq n \leq 10^5, 1 \leq k \leq 10^5, 1 \leq n * k \leq 10^7$), oznaczające liczbę książek i długość niedokońzonego programu.

W drugim wierszu znajduje się słowo składające się z k znaków U, D i ?, oznaczające niedokończony program.

W trzecim wierszu znajduje się n liczb a_i , oznaczających numery kolejnych książek na półce.

Wyjście

W jedynym wierszu wyjścia należy wypisać n liczb całkowitych, gdzie i -ta ($0 \leq i \leq n-1$) z nich oznacza liczbę dokończeń programu dających wynik i , modulo 998244353.

Przykłady

Wejście dla testu r4e0a:

```
7 3
?DU
3 1 6 7 5 4 2
```

Wyjście dla testu r4e0a:

```
0 0 2 0 0 0 0
```

Wyjaśnienie: Mamy dwa możliwe dokończenia: DDU i UDU, gdzie w obu przypadkach trzeba uruchomić program 2 razy. Np. przy dokończeniu DDU ciąg będzie się zmieniał w następujący sposób:

Uruchomienie nr 1
 $3, 1, 6, 7, 5, 4, 2 \xrightarrow{D} 1, 3, 2, 6, 7, 5, 4 \xrightarrow{D} 1, 2, 3, 4, 6, 7, 5 \xrightarrow{U} 1, 2, 3, 4, 6, 5, 7$

Uruchomienie nr 2
 $1, 2, 3, 5, 6, 5, 7 \xrightarrow{D} 1, 2, 3, 4, 5, 6, 7 \xrightarrow{D} 1, 2, 3, 4, 5, 6, 7 \xrightarrow{U} 1, 2, 3, 4, 5, 6, 7$

Zamiatanie

Limit pamięci: 256 MB

Wejście dla testu r4e0b:

```
7 3
???
3 1 6 7 5 4 2
```

Wyjście dla testu r4e0b:

```
0 1 7 0 0 0 0
```

Wejście dla testu r4e0c:

```
8 2
??
8 3 6 5 2 4 7 1
```

Wyjście dla testu r4e0c:

```
0 0 0 2 2 0 0 0
```

Wejście dla testu r4e0d:

```
16 5
?U???
16 7 15 3 9 8 4 6 11 13 2 5 14 10 12 1
```

Wyjście dla testu r4e0d:

```
0 0 15 1 0 0 0 0 0 0 0 0 0 0 0 0
```

Ocenianie

Podzadanie	Ograniczenia	Limit czasu	Punkty
1	$n \leq 10, k \leq 3$	1 s (C++) / 7 s (Python)	1
2	$n \leq 1000, k \leq 3$	2 s (C++) / 7s (Python)	12
3	Program jest już dokończony i nie zawiera znaków ?	1 s (C++) / 10 s (Python)	20
4	$n \leq 400, k \leq 400$	3 s (C++) / 50 s (Python)	25
5	Brak dodatkowych ograniczeń	11 s (C++) / 120 s (Python)	42