

# Zadanie: HAK

## Hanoj kontratakuję



XXXIV OI, etap I. Plik źródłowy hak.\* Dostępna pamięć: 256 MB.

12.10–16.11.2026

Uwaga: W tym zadaniu poznasz wynik punktowy swoich zgłoszeń **dopiero po zakończeniu zawodów**.

Bajtosią zaczęła niedawno naukę w liceum w Bajtosinie i poznała Bajtkę. Aby lepiej się z nią zaprzyjaźnić, zaprosiła go do wspólnej gry w *Wieżę Hanoj*. Mimo podobnie brzmiącej nazwy, jest to zupełnie inna gra niż *Wieża Hanoi*! W *Wieżach Hanoj* mamy do dyspozycji  $k$  wież oraz  $n$  krążków o parami różnych rozmiarach ponumerowanych od 1 do  $n$  w kolejności od najmniejszego do największego. Krążki mogą leżeć na wieżach w dowolnej kolejności, niezależnie od swoich rozmiarów. W jednym ruchu gracz może zdjąć krążek z wierzchu dowolnej wieży i położyć go **na sam dół** (pod wszystkie znajdujące się tam krążki) wybranej wieży. W szczególności dozwolone jest zdjęcie krążka z góry wieży i włożenie go na spód tej samej wieży.

W pewnym momencie Bajtosię zawołała jej mama, więc dziewczyna wyszła z pokoju. Pod jej nieobecność Bajtek zaczął się nudzić, więc wziął zabawkę i zaczął przekładać krążki zgodnie z zasadami gry. Po chwili zorientował się jednak, że całkowicie zmienił układ zabawki i nie wie, jak przywrócić go do stanu, w którym przerwali zabawę.

Twoim zadaniem jest stworzyć instrukcję dla Bajtki, dzięki której będzie mógł przywrócić zabawkę Bajtosi do stanu pierwotnego. Bajtkowi zależy na czasie (Bajtosi może wrócić w każdej chwili), więc bardzo chciałby, aby instrukcja zawierała co najwyżej 4 000 000 ruchów. Możesz założyć, że zawsze jest to możliwe.

## Wejście

W pierwszym wierszu znajduje się jedna liczba całkowita  $k$  oznaczająca liczbę wież ( $2 \leq k \leq 5$ ).

W następnych  $k$  wierszach znajduje się opis konfiguracji pierwotnej, czyli układu krążków z momentu wyjścia Bajtosi z pokoju, który należy odtworzyć. W  $i$ -tym ( $1 \leq i \leq k$ ) wierszu opisu konfiguracji pierwotnej znajduje się liczba całkowita  $c_i$  ( $0 \leq c_i \leq n$ ) oznaczająca liczbę krążków na wieży numer  $i$ . Następnie znajduje się  $c_i$  dodatknych liczb całkowitych  $p_{i,1}, p_{i,2}, \dots, p_{i,c_i}$ , oddzielonych spacjami, oznaczających krążki na  $i$ -tej wieży. Krążki są podane w kolejności od góry do dołu. Wartości  $p_{i,j}$  (dla  $1 \leq j \leq c_i$ ) należą do zakresu od 1 do  $n$ . Każdy z  $n$  krążków znajduje się na dokładnie jednej wieży w konfiguracji pierwotnej.

W następnych  $k$  wierszach znajduje się opis konfiguracji aktualnej, czyli układu krążków po zabawie Bajtki, przedstawiony w analogiczny sposób. W  $i$ -tym ( $1 \leq i \leq k$ ) wierszu opisu konfiguracji aktualnej znajduje się liczba całkowita  $d_i$  ( $0 \leq d_i \leq n$ ) oznaczająca liczbę krążków na wieży numer  $i$ . Następnie znajduje się  $d_i$  dodatknych liczb całkowitych  $k_{i,1}, k_{i,2}, \dots, k_{i,d_i}$ , oddzielonych spacjami, oznaczających krążki na  $i$ -tej wieży. Krążki są podane w kolejności od góry do dołu. Wartości  $k_{i,j}$  (dla  $1 \leq j \leq d_i$ ) należą do zakresu od 1 do  $n$ . Każdy z  $n$  krążków znajduje się na dokładnie jednej wieży w konfiguracji aktualnej.

W każdej z obu konfiguracji znajduje się łącznie  $n$  krążków. Gwarantujemy, że  $1 \leq n \leq 100\,000$ .

## Wyjście

W pierwszym wierszu wyjścia Twój program powinien wypisać jedną liczbę całkowitą  $m$  ( $0 \leq m \leq 4\,000\,000$ ) oznaczającą liczbę ruchów, które powinien wykonać Bajtek, aby przywrócić pierwotny stan *Wież Hanoj*. W następnych  $m$  wierszach powinny znaleźć się opisy kolejnych ruchów, które powinien wykonać Bajtek. Każdy opis składa się z dwóch liczb całkowitych  $a$  oraz  $b$  ( $1 \leq a, b \leq k$ ), oddzielonych spacją, oznaczających, że Bajtek powinien wziąć krążek z góry wieży  $a$  i przenieść go na spód wieży  $b$ . W chwili wykonywania ruchu wieża  $a$  musi być niepusta. Aby wyjście Twojego programu zostało uznane za poprawne, musi składać się z maksymalnie 4 000 000 ruchów, a po ich wykonaniu (zaczynając od konfiguracji aktualnej) konfiguracja musi być identyczna z konfiguracją pierwotną. Jeśli istnieje wiele poprawnych instrukcji, Twój program może wypisać dowolną z nich.

## Przykład

Dla danych wejściowych:

2  
0  
2 1 2  
2 2 1  
0

jednym z poprawnych wyników jest:

3  
1 1  
1 2  
1 2

**Wyjaśnienie przykładu:** W pierwszym przykładzie w konfiguracji aktualnej krążki na pierwszej wieży leżą w kolejności 2, 1 od góry do dołu, a druga wieża jest pusta. Przekładamy krążek 2 na spód tej samej wieży, dzięki czemu krążek 1 znajduje się na wierzchu. Następnie przenosimy oba krążki na drugą wieżę.

Natomiast dla danych wejściowych:

3  
2 2 4  
1 1  
1 3  
2 1 2  
1 3  
1 4

jednym z poprawnych wyników jest:

5  
1 2  
2 2  
3 1  
2 2  
2 3

**Wyjaśnienie przykładu:** W drugim przykładzie w konfiguracji aktualnej wieże zawierają kolejno (1, 2), (3) i (4), a mają zawierać (2, 4), (1) i (3). Przenosimy krążek 1 na drugą wieżę, pod krążek 3. Przekładamy krążek 3 na spód tej samej wieży, odsłaniając krążek 1. Krążek 4 dokładamy na spód pierwszej wieży. Potem przekładamy krążek 1 na spód drugiej wieży, dzięki czemu krążek 3 można przenieść na trzecią wieżę.

**Testy przykładowe:** Testy 0a, 0b to testy przykładowe powyżej. Pozostałe testy przykładowe mają następujące konfiguracje. Krążki podano w kolejności od góry do dołu:

**0c:**  $n = 100$ ,  $k = 5$ . Dla  $j = 1, \dots, 5$  niech  $I_j$  oznacza kolejno przedziały  $[1, 60]$ ,  $[61, 80]$ ,  $[81, 90]$ ,  $[91, 97]$ ,  $[98, 100]$ . W konfiguracji aktualnej krążki na wieży  $j$  mają rozmiary  $t^7 \bmod 101$  dla kolejnych  $t \in I_j$ ; w konfiguracji pierwotnej mają rozmiary  $t^{13} \bmod 101$  dla kolejnych  $t \in I_j$ .

**0d:**  $n = 2000$ ,  $k = 5$ . W konfiguracji aktualnej wieża  $i$  zawiera kolejno krążki o rozmiarach  $400(i-1) + 1$ ,  $400(i-1) + 2, \dots, 400i$ . W konfiguracji pierwotnej zawiera kolejno krążki o rozmiarach  $i, i+5, i+10, \dots$  nie większych niż  $n$  ( $i = 1, \dots, 5$ ).

**0e:**  $n = 100\,000$ ,  $k = 2$ . W konfiguracji aktualnej wieża 1 zawiera kolejno krążki o rozmiarach 100 000, 99 999,  $\dots$ , 1, a w konfiguracji pierwotnej 1, 2,  $\dots$ , 100 000. Wieża 2 jest pusta w obu konfiguracjach.

## Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów. W każdym podzadaniu istnieją grupy testów warte 80% punktów za to podzadanie, w których spełniony jest warunek  $k = 5$ .

| Podzadanie | Dodatkowe ograniczenia      | Punkty |
|------------|-----------------------------|--------|
| 1          | $n \leq 100$                | 10     |
| 2          | $n \leq 2\,000$             | 30     |
| 3          | brak dodatkowych ograniczeń | 60     |