Zadanie: TET
Tetris Attack

Etap II. Dzień drugi. Plik źródłowy `tet.*` *08.02.2007*
Dostępna pamięć: 32 MB.

Ostatnimi czasy w Bajtocji bardzo popularną grą stała się łamigłówka „Tetris attack". Jej uproszczona wersja ma następującą postać: Gracz otrzymuje do dyspozycji stos, na którym umieszczono $2n$ elementów (jeden na drugim), oznaczonych n różnymi symbolami. Przy tym każdym symbolem są oznaczone dokładnie dwa elementy na stosie. Ruch gracza polega na zamianie dwóch sąsiednich elementów miejscami. Jeśli w wyniku zamiany na stosie sąsiadują ze sobą elementy oznaczone identycznymi symbolami, to w „magiczny" sposób znikają, a elementy znajdujące się powyżej spadają w dół (być może powodując kolejne zniknięcia). Celem gracza jest opróżnienie stosu w jak najmniejszej liczbie ruchów.

Zadanie

Napisz program który:

- wczyta ze standardowego wejścia opis początkowej zawartości stosu,
- obliczy rozwiązanie wymagające minimalnej liczby ruchów,
- wypisze znalezione rozwiązanie na standardowe wyjście.

Wejście

W pierwszym wierszu standardowego wejścia znajduje się jedna liczba całkowita n, $1 \leq n \leq 50\,000$. W kolejnych $2n$ wierszach zapisana jest początkowa zawartość stosu. Wiersz $i+1$-wszy zawiera jedną liczbę całkowitą a_i — symbol elementu znajdującego się na wysokości i ($1 \leq a_i \leq n$). Każdy symbol występuje na stosie **dokładnie** 2 razy. Na początku żadne dwa identyczne symbole nie występują obok siebie. Ponadto dane testowe są tak dobrane, że istnieje rozwiązanie zawierające nie więcej niż 1 000 000 ruchów.

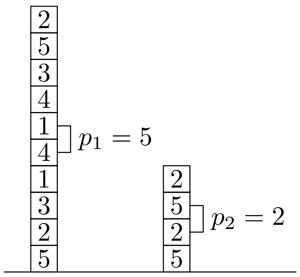
Wyjście

Na standardowym wyjściu należy wypisać opis rozwiązania, wymagającego minimalnej liczby ruchów. Pierwszy wiersz powinien zawierać jedną liczbę całkowitą m — długość najkrótszego rozwiązania. Kolejne m wierszy powinno zawierać opis rozwiązania, czyli ciąg m liczb całkowitych $p_1, \ldots, p_m$, po jednej w każdym wierszu. Wartość p_i oznacza, że w i-tym ruchu gracz zdecydował o zamianie elementów, znajdujących się na wysokościach p_i oraz $p_i + 1$.

Jeżeli istnieje wiele rozwiązań, to Twój program powinien wypisać dowolne z nich.

Przykład

Dla danych wejściowych:

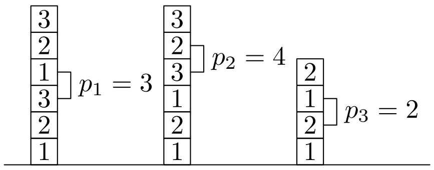


5
5
2
3
1
4
1
4
3
5
2

poprawnym wynikiem jest:
2
5
2

Natomiast dla danych wejściowych:
3
1
2
3
1
2
3

poprawnym wynikiem jest:
3
3
4
2



jak również:
3
4
3
2