

Zadanie: KLO

Klocki



XXI OI, etap I. Plik źródłowy klo.* Dostępna pamięć: 64 MB.

7.10–4.11.2013

Mały Bitek i jego koledzy cały wczorajszy dzień w przedszkolu spędzili na zabawie kolorowymi klockami. Tworzyli przeróżne budowle, ale szybko im się to znudziło. Postanowili więc poustawiać klocki jeden za drugim. Unikali oni przy tym sytuacji, w której pewien klocek stoi tuż przed klockiem tego samego koloru. Po dłuższym czasie udało im się ustawić w ten sposób wszystkie klocki. Wtedy zajęcia się skończyły i rodzice odebrali dzieci z przedszkola.

Dzisiaj Bitek przybył do przedszkola pół godziny przed zajęciami. Zobaczył, że dzieło zbudowane poprzedniego dnia nadal istniało. Niestety, Bitek przewrócił się i wszystkie klocki się wymieszały. Chłopiec natychmiast uporządkował klocki i zaczął zastanawiać się, jak szybko odbudować to, co zepsuł. Przypomniawszy sobie, jakiego koloru klocki stały na końcach rzędu.

Pomóż małemu Bitkowi i podpowiedz, jak może ustawić klocki, żeby żaden klocek nie stał przed klockiem tego samego koloru oraz żeby klocki na końcach miały takie kolory, jakie podał Bitek. Jeśli Bitkowi coś się pomyliło lub chłopiec przypadkowo zagubił niektóre klocki podczas upadku, tak że teraz ustawienie zgodne z jego warunkami nie jest możliwe, pomóż mu to stwierdzić.

Wejście

W pierwszym wejściu standardowego wejścia znajdują się trzy liczby całkowite k , p i q ($1 \leq k \leq 1\,000\,000$, $1 \leq p, q \leq k$), pooddzielane pojedynczymi odstępami i oznaczające liczbę kolorów klocków oraz kolory pierwszego i ostatniego klocka w szukanym ustawieniu. W drugim wierszu znajduje się k liczb całkowitych i_1, i_2, \dots, i_k , pooddzielanych pojedynczymi odstępami ($1 \leq i_j \leq 1\,000\,000$). Liczba i_j oznacza, że Bitek ma dokładnie i_j klocków koloru j . Możesz założyć, że liczba wszystkich klocków nie przekracza miliona, tzn. $n = i_1 + i_2 + \dots + i_k \leq 1\,000\,000$.

Wyjście

Twój program powinien wypisać na standardowe wyjście n liczb całkowitych pooddzielanych pojedynczymi odstępami reprezentujących kolory kolejnych klocków w szukanym ułożeniu. Jeśli takie ułożenie nie istnieje, Twój program powinien wypisać tylko jedną liczbę całkowitą: 0.

Jeśli jest wiele poprawnych odpowiedzi, Twój program może podać dowolną z nich.

Przykład

Dla danych wejściowych:

3 3 1
2 3 3

jednym z poprawnych wyników jest:

3 2 1 3 2 3 2 1

a dla danych wejściowych:

3 3 1
2 4 2

poprawnym wynikiem jest:

0

Wyjaśnienie do przykładów:

Innym poprawnym ułożeniem klocków w pierwszym przykładzie jest 3 1 2 3 2 3 2 1. W drugim przykładzie Bitek musiał się pomylić – nie można ułożyć klocków tak, żeby spełnione były warunki z treści zadania.