Napisz program w Javie, który scala $n$ posortowanych ciągów liczb całkowitych, których długości nie są większe od $m$, działający w czasie **O(m*nlogn)** i wykorzystujący dodatkową pamięć **O(n)**.

**Wejście**

Dane do programu wczytywane są ze standardowego wejścia (klawiatury) zgodnie z poniższą specyfikacją:

1. Pierwszą podawaną wartością będzie dodatnia liczba całkowita $z$ ($1 \leq z \leq 100$), oznaczająca ilość zestawów danych.
2. Każdy zestaw danych ma następującą postać:
   a. W pierwszej linii znajduje się liczba całkowita $n$ ($1 \leq n \leq 1000$), oznaczająca liczbę ciągów.
   b. W następnej linii zapisanych jest $n$ liczb: $d_1, d_2, \ldots, d_n$, oznaczających długości scalanych ciągów, przy czym: $1 \leq d_i \leq m$, $m = max\{d_i, i=1, \ldots, n\}$ przy czym ($1 \leq m \leq 1000$).
   c. W kolejnych $n$ liniach znajdują się uporządkowane niemalejąco liczby typu `int`, reprezentujące scalane ciągi.

**Wyjście**

Dla każdego zestawu danych wypisz w jednej linii tablicę zawierającą elementy wszystkich ciągów podanych na wejściu, przy czym po każdym elemencie ciągu występuje znak spacji.

**Wymagania implementacyjne**

Jedynym możliwym importem jest **`java.util.Scanner`**.

**Przykład.**

| **Wejście:** | **Wyjście:** |
|---|---|
| 2<br>4<br>1 7 3 10<br>0<br>1 3 5 7 9 11 13<br>2 4 6<br>1 2 3 4 5 6 7 8 9 10<br>3<br>7 8 10<br>4 4 4 4 4 4 4<br>1 3 5 7 9 11 13 15<br>1 2 3 4 5 6 7 8 9 10 | 0 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 9 9 10 11 13<br>1 1 2 3 3 4 4 4 4 4 4 4 4 5 5 6 7 7 8 9 9 10 11 13 15 |