Opis

Dana jest niepusta tablica dwuwymiarowa liczb całkowitych o wymiarach n x m (n i m są liczbami naturalnymi większymi od 0). Tablica ta ma każdy wiersz i każdą kolumnę posortowaną niemalejąco. Napisz w Javie program wyszukujący podany element w tablicy za pomocą czterech różnych funkcji:

- funkcji rekurencyjnej, wyznaczającej pierwsze wystąpienie elementu,
- funkcji rekurencyjnej, wyznaczającej ostatnie wystąpienie elementu,
- funkcji iteracyjnej, wyznaczającej pierwsze wystąpienie elementu,
- funkcji iteracyjnej, wyznaczającej ostatnie wystąpienie elementu.

Wszystkie funkcje mają mieć średnią złożoność czasową **silnie mniejszą** od złożoności kwadratowej – czyli mniej niż $O(\max(n,m))^2$.

Pierwsze wystąpienie elementu oznacza najwcześniejsze wystąpienie leksykograficznie (przede wszystkim jak najwcześniejszy wiersz, następnie jak najwcześniejsza kolumna), natomiast ostatnie – najpóźniejsze leksykograficznie. W ciele funkcji rekurencyjnych nie może pojawić się żaden rodzaj pętli, natomiast funkcje iteracyjne nie mogą wywoływać innej utworzonej funkcji.

Wejście

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

Pierwszą podawaną wartością będzie dodatnia liczba całkowita oznaczająca ilość zestawów danych, po której na wejściu pojawią się zestawy danych w ilości równej wczytanej liczbie.

Każdy zestaw danych zawiera dwie dodatnie liczby całkowite z zakresu od 1 do 100, oznaczające odpowiednio liczbę wierszy oraz liczbę kolumn tablicy, w następnych liniach podawane są dane będące kolejnymi wierszami tablicy zgodnie z podaną liczbą wierszy i kolumn. Dane każdego zestawu są liczbami całkowitymi z zakresu od -2^{15} do $+2^{15}$.

Na samym końcu podawana jest wartość, która ma zostać wyszukana w tabeli kolejno za pomocą wszystkich czterech funkcji.

Wyjście

Dla każdego zestawu danych program wypisuje cztery linie, będące efektem działania kolejnych czterech funkcji oraz znak „---” w piątej:

RekPier: k w (i1,j1)

RekOst: k w (i2,j2)

IterPier: k w (i3,j3)

IterOst: k w (i4,j4)

Litery przed dwukropkiem oznaczają identyfikator wywołanej funkcji, k oznacza szukany element, natomiast kolejne indeksy i oraz j oznaczają wiersz i kolumnę ze znalezionym wystąpieniem element. W przypadku braku elementu w tablicy, po dwukropku ma pojawić się napis „ **nie ma k**”, gdzie k jest szukanym elementem (patrz przykład)

Wymagania implementacyjne

1. W pierwszej linii program powinien zawierać komentarz: **// Nazwisko i imię – nr grupy**
2. Jedynym dozwolonym importem jest obsługa wczytywania z klawiatury, to jest: **import java.util.Scanner;**
3. Główna klasa musi nazywać się **Source**, co oznacza ogólne ramy kodu postaci:

```
class Source {
  public static void main( String [] args ) {
  …   }
}
```

4. Wczytywanie musi się odbywać przez pojedynczą zmienną skanera wczytywania, zadeklarowaną zewnętrznie w stosunku do wszystkich metod głównej klasy.

   W praktyce oznacza to tylko jedną deklarację w przykładowej postaci:

   public static Scanner in = new Scanner(System.in);

   w pierwszej linii ciała głównej klasy.

5. Na końcu pliku źródłowego proszę podać w komentarzu przykłady własnych danych testowych, sprawdzających wszystkie przypadki występujące w programie, dla których był uruchomiony program przed wysłaniem na BaCę.

Przykład danych

Wejście:	**Wyjście:**
3 3 4 10 10 10 10 10 20 20 30 20 20 20 40 20 3 4 10 10 10 10 10 20 20 30 20 20 20 40 50 3 3 10 10 10 10 20 20 20 20 20 10	RekPier: 20 w (1,1) RekOst: 20 w (2,2) IterPier: 20 w (1,1) IterOst: 20 w (2,2) --- RekPier: nie ma 50 RekOst: nie ma 50 IterPier: nie ma 50 IterOst: nie ma 50 --- RekPier: 10 w (0,0) RekOst: 10 w (1,0) IterPier: 10 w (0,0) IterOst: 10 w (1,0) ---

Metody programowania 2019/2020
Wyszukiwanie w tablicy 2D

P_06