Problem 1

Define a two-dimensional, rectangular array of **int**s and then calculate and print the sum of all elements from the last column. No modifications of your program should be necessary when the array (in particular its dimensions) is modified.

Problem 2

Define in your program a two-dimensional rectangular array of **int**s, for example

```
int[][] arr = { {1,3,2}, {3,4,8}, {2,6,8}, {1,8,5} };
```

and then calculate the index of the 'row' with the largest sum of elements and the index of the 'column' with the largest sum of elements (in the example above this would be indices 2 and 2). Of course, the program should work for any two-dimensional rectagular array, with arbitrary dimensions.

Problem 3

Define a two-dimensional array of **String**s with names of countries and their capitals, for example as below

download *Arr2DCapitals.java*

```
String[][] arr =
      { {"Kenya",  "Nairobi"}, {"Rwanda", "Kigali"},
        {"Gambia", "Banjul"},  {"Ghana",  "Accra"},
        {"Niger",  "Niamey"},  {"Zambia", "Lusaka"} };
```

and then write a program which reads the name of a country and prints the name of its capital (don't forget that **String**s should be compared using **equals** or **equalsIgnoreCase**, *not* == operator!). The program should display an appropriate message if the country name entered cannot be found.

Problem 4

Define two-dimensional array of **int**s and two indices. The program should exchange (swap) 'rows' of the array specified by the two indices. Print the array before and after the operations. The array doesn't have to be rectangular; do *not* create any auxiliary arrays.

Problem 5

Define a three-dimensional array of **int**s representing deposits and withdrawals of customers of a bank in and out of their accounts (each customer may have several accounts). For example

download *Arr3DBank.java*

```
int[][][] opers = {
    { {100, -50, 25}, {150,-300}, {300,-90,100} },
```

```
        { {90, -60, 250}, {300,20,-100} },
        { {20, 50}, {300}, {20,-20,40}, {100,-200} }
    };
```

where

- the first index indicates a customer;
- the second index indicates, for a given customer, his/her account;
- the third index indicates, for a given customer and his/her account, subsequent deposits (positive values) and withdrawals (negative values).

The program should create an array of **int**s of dimension equal to the number of customers, the elements of which are sums of all deposits and withdrawals for subsequent customers, in and out of all his/her accounts (for the data as in the example above, these should be the numbers 235, 500 and 310).

After modifications of the sizes and/or values of elements of the array oper, the program should work correctly without any other changes.
