

Programming 1 – Exam 3

August 23, 2021

Submit the files `Prva.java`, `Druga.java`, `Tretja.java`, and `Cetrta.java`. You can test them as follows:

(1) `tj.exe Prva.java . .` (2) `tj.exe` (3) `tj.exe` (4) `tj.exe`

- ① The input consists of integers $d \in [1, 6]$ and $v \in [1, 9d]$, separated by a space. Write a program that prints the smallest d -digit integer whose sum of digits is v .

Test case (input/output):

4 23

1499

- ② Miha bought an NNNNN (Nekoliko nenavaden nosilec nadležne navlake; Somewhat Unusual Carrier of Annoying Clutter), a piece of furniture divided into m “rows” with heights h_1, h_2, \dots, h_m and n “columns” with widths w_1, w_2, \dots, w_n , forming $m \times n$ compartments. Miha will use it to store as many of his k rectangular items as possible; the sizes of the items are $a_1 \times b_1, a_2 \times b_2, \dots, a_k \times b_k$. To this end, he traverses the compartments in the usual fashion (from the top row to the bottom row, and in the left-to-right direction within each row) and, for each compartment, chooses an item to be stored in it. This is the item with the greatest area that still fits into the current compartment; if there are multiple such items, Miha picks the first of them. Each item may be rotated by 90 degrees. Miha’s procedure is over when none of the remaining items can be placed into the current compartment. In the example shown below, Miha can store three items into the NNNNN, since none of the remaining items can be placed into the first compartment in the second row.

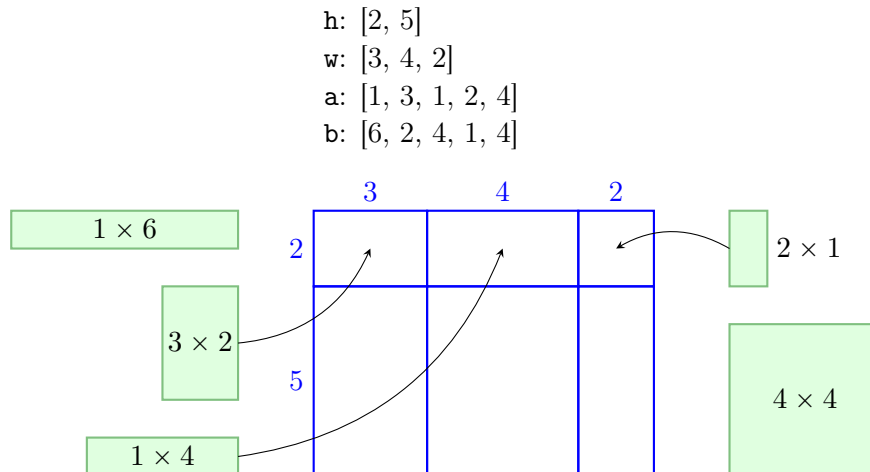
Complete the following method in the class `Druga`:

```
public static int steviloShranjenih(int[] h, int[] w, int[] a, int[] b)
```

The method returns the number of items that Miha can store into his NNNNN using the given procedure. The array `h` holds the integers h_i , the array `w` holds the integers w_i , etc. Each array contains at least one and at most 200 elements, and all elements belong to the interval $[1, 1000]$. The arrays `a` and `b` have the same length.

In the method, you may modify the elements of the input arrays if desired.

In 60% of the hidden test cases, all items are square-shaped. In a half of them, all items have the same size.



- ③ The following classes are defined as static inner classes in the class `Tretja`:

```
abstract class Zival {}           // Animal
class Pes extends Zival {}       // Dog
class Macka extends Zival {}     // Cat
class RodovniskiPes extends Pes { // Purebred dog
    private String pasma;         // breed
    private RodovniskiPes oce;    // father;
                                // null if this is the beginner of the family
    private RodovniskiPes mati;  // mother;
                                // null if this is the beginner of the family
    // in all test cases: oce == null  $\iff$  mati == null
}
```

Do the following tasks:

- [34%] Complete the following method in the class `Tretja`:

```
public static int prestej(RodovniskiPes[] psi, String pasma)
```

The method returns the number of dogs of the breed `pasma` in the array `psi`.

- [34%] Complete the classes in such a way that the method

```
public static void koncert(Zival[] zivali) {
    for (Zival zival: zivali) {
        System.out.println(zival.oglasise());
    }
}
```

will print `hov` for every dog and `mijav` for every cat.

- [32%] Complete the following method in the class `RodovniskiPes`:

```
public boolean preveri()
```

The method returns `true` if and only if one of the following holds: (1) `this` dog is the beginner of the family; (2) the father and the mother are both of the same breed as `this` dog, and both of them satisfy condition (1) or (2).

- ④ The class `Cetrta` contains the following static inner class:

```
class Prostor {    // Room
    private int nadstropje;    // floor number (i.e., level)
    private int kvadratura;    // area
    private String namembnost; // purpose, e.g., "pisarna" (office),
                                // "kuhinja" (kitchen) ...
}
```

Complete the following methods (you may also add your code to the class `Prostor` if needed):

- [34%] `public static Set<String> namembnosti(List<Prostor> prostori)`

Returns the set of all purposes of the rooms in the given list.

- [34%] `public static void uredi(List<Prostor> prostori)`

Sorts the given list of rooms by increasing floor number. The rooms on the same floor have to be sorted by decreasing area.

- [32%] `public static Map<Integer, Map<String, Integer>> statistika(List<Prostor> prostori)`

Returns a dictionary that maps an integer n to a dictionary that maps a string s to the number of rooms from the list `prostori` that are located on the n -th floor and whose purpose is s .