

## Zadání úkolu z 10. cvičení

### Úlohy

1. Zapište program `PocetyCifer`, který bude porovnávat, zda pro dvojici zadaných celých čísel (načtených do proměnných typu `long`) platí, že jejich zápis je tvořen stejnými ciframi (stejně cifry ve stejném počtu).

Program má opakovaně načítat dvojice celých kladných čísel, dokud na místě prvního z nich není zadané číslo záporné. Po zadání záporného program ukončit.

Pro každou dvojici zadaných čísel musí pak program zjistit počty cifer každého z čísel a následně tyto počty navzájem porovnat. Pro tyto účely je možné/vhodné vytvořit dvě metody, kde první by mohla zjistit počty cifer jednoho čísla a výsledek vrátit v poli, druhá pak porovnat dvě pole navzájem.

2. Zapište program `VycetkaPlatidel`, který pro zadané celočíselné hodnoty (představující finanční částku například v korunách) vždy vytvoří a vypíše výčetku platidel, tj. kolik bankovek a mincí je třeba pro zaplacení dané částky.

Program má načítat čísla do zadání nuly nebo záporného a pro každé zadané číslo pak ve vhodném tvaru vypsat výčetku.

Hodnoty jednotlivých platidel budou v programu uchovány v poli, které vytvoříte pomocí statického inicializátoru např.: `int[] platidla = {5000, 2000, 1000, 500, 200, 100, 50, 20, 10, 5, 2, 1};`

3. Zapište program `PocetyHodnotVIntervalech`, který bude určovat počty zadaných hodnot v určených intervalech. Předpokládejme, že máme k dispozici  $n$  vzestupně uspořádaných čísel/hodnot  $m_1, m_2, m_3 \dots m_n$ . Pomocí těchto hodnot můžeme množinu reálných hodnot rozdělit do  $n+1$  intervalů  $(-\infty, m_1>, (m_1, m_2> \dots (m_n, \infty)$ . Dále jsou máme sadu dat (hodnot  $x_i$ ), program má určit kolik z nich je v tom kterém intervalu.

Data pro zpracování program načte ze standardního vstupu. Na vstupu bude nejprve  $n$  potom budou následovat meze intervalů  $m_1, m_2, m_3 \dots m_n$  a poté jsou zadávána data/čísla sady.

Meze intervalů  $m_1, m_2, m_3 \dots m_n$  po/při načtení uložte do vhodného pole. Naopak čísla zpracovávané sady zpracovávejte pouze sekvenčně nenačítejte je do pole. Jednotlivé hodnoty sady načítejte do zadání libovolného jiného textového řetězce, než odpovídá formátu čísla, tj.

```
while (sc.hasNextDouble()) {
    cislo = sc.nextDouble();
    // zpracovat cislo, navyšit počet v odpovídajícím intervalu
}
// konec zpracování, výpis výsledků tj počty hodnot v jednotlivých
intervalech
```

4. Zapište program `EratosthenovoSito`, který zjistí všechna prvočísla do zadané maximální hodnoty metodou *Eratosthenova síta*. Program na vstupu obdrží celé kladné číslo  $n$ , následně má na výstup vypsat všechna prvočísla od 2 do  $n$ . Opět program může pracovat opakovaně například do zadání hodnoty menší 2.

Eratosthenovo síto je pole s indexy do  $n$  (pole velikosti  $n+1$ ) hodnot typu `boolean`. Indexy do pole jsou čísla, jejichž prvočíselnost potřebujeme zjistit. Pole používáme v rozsahu indexů od 2 do  $n$  ( $n$  odpovídá délce pole). Před zahájením algoritmu jsou v poli v celém rozsahu hodnoty `true`. O všech číslech „předpokládáme“ prvočíselnost. Postupně pak budeme označovat ta čísla, která tuto vlastnost nemají, tj. pro všechny prvky v poli od indexu 2, pokud je příslušné číslo prvočíslem označíme si všechny jeho násobky jako „neprvočíselné“, tedy do příslušné pozice v poli přiřadíme `false`.