

Vše si pamatuji, mám to v paměti (poli)

Načtení dat do pole. Přístup k datům v poli.

Úlohy

Pole, algoritmická primitiva, metody

1. Zapište program pro načtení číselných hodnot ze standardního vstupu do pole. Vstup realizujte postupně ve dvou variantách: v první variantě bude na vstupu nejprve zadán celkový počet hodnot a následně hodnoty; ve druhé variantě budou načítána čísla do zadání předem určené koncové hodnoty.
2. Vytvořte třídu `ArrayTools`, ve které budou metody pro některé základní operace s polem čísel:
 - a) Metoda pro výpočet součtu hodnot v poli.
 - b) Metoda pro výpočet součinu hodnot v poli.
 - c) Metoda pro nalezení maximální hodnoty
 - d) Metoda pro zjištění minimální hodnoty
 - e) Metody pro zjištění první (variantně i poslední) pozice výskytu maxima a minima.
 - f) Metoda pro zjištění počtu výskytů zadané hodnoty.
 - g) Metoda pro zjištění počtu sudých/lichých.
 - h) Metody pro zjištění prvního a posledního výskytu zadané hodnoty.
 - i) Metody pro testování vzestupné a sestupné uspořádanosti hodnot v poli.
 - j) Metoda pro reverzi hodnot v poli.
 - k) Metoda pro generování pole zadané délky s náhodnými hodnotami v zadaném rozsahu.

Metody dle bodů a)–j) vytvořte ve dvou variantách – v první variantě pro libovolně dlouhé pole s n hodnotami od indexu 0 do indexu $n-1$ a v druhé variantě pro zcela naplněné pole. Při zápisu kódu druhé varianty metody využijte variantu první.

Zároveň s implementací jednotlivých metod realizujte základní testování metod (testování provedte např. s polem vytvořeným pomocí statického inicializátoru, nebo náhodně vytvořeným polem).

Některé z výše uvedených bodů a)–k) realizujte i pro pole reálných čísel.

3. Zapište program pro „uživatelské“ testování (některých) prostředků implementovaných ve třídě `ArrayTools`. Program má uživateli umožňovat opakované zadávání sekvencí čísel. Každá další sekvence bude zadaná počtem, následně hodnotami. Program má skončit svoji činnost v případě zadání záporného počtu, nulový počet jsou platná data pro prázdnou sekvenci. Každou zadanou sekvenci vždy načtěte do pole. Dále na toto pole aplikujte funkce/metody deklarované ve třídě `ArrayTools` – respektive ty metody, které jsou na pole daného typu aplikovatelné. Výsledky činnosti každé metody vypište uživateli.
4. Zapište program obdobně jako v předchozí úloze s tím rozdílem, že v tomto případě bude program načítat posloupnosti čísel ukončené „vhodným terminátorem“. Omezte se na zpracování posloupností nezáporných čísel – zadáním záporné hodnoty bude uživatel indikovat „konec posloupnosti“. Opět vytvořte program tak, aby umožňovat opakované řešení (zadání a zpracování posloupnosti) a to v tomto případě na základě explicitního vyjádření uživatele – tj. uživateli vypsát dotaz např. „Pokračovat řešením další úlohy?“, pak po zadání `n/ne/no` (malými či velkými písmeny) program ukončit.

Mám to po ruce

5. Zapište program, který zjistí a vypíše všechna prvočísla od 2 do k . Pro zjištění prvočísel v zadaném rozsahu použijte metodu Eratosthenova síta. Hodnota k je omezena maximální velikostí alokovaného pole.
6. Uvažujte dále variantu algoritmu, ve kterém budete každé číslo zvlášť testovat na prvočíselnost opakovaným hledáním dělitelů. Porovnejte oba algoritmy co do počtu prováděných „jednotkových operací“.

7. Zapište a otestujte metodu, které bude zjišťovat a poskytovat počty cifer pro hodnoty typu `long` předané jako parametr. Metodu použijte v programu, který bude testovat, zda zadaná čísla obsahují stejné cifry (zadaná čísla mohou být 2, může jich být ale i libovolně dlouhá sekvence).
8. Zapište program pro otestování generátoru náhodných čísel. Program nechť načte dvě celá kladná čísla `rozsah` a `pocet`. Program má následně vygenerovat `pocet` čísel v rozsahu od 0 do `rozsah - 1` a zjistit (a následně vypsat) počty jednotlivých vygenerovaných čísel.

Zjistěte minimální a maximální počet výskytů apod.

Pro dostatečně velký `pocet` (vzhledem k zadané hodnotě `rozsah`) by počty jednotlivých vygenerovaných hodnot měly být „skoro“ stejné.

9. Zapište program, který pro zadanou sumu v korunách vypíše výčetku platidel – tj. počet a hodnoty bankovek, kterými lze zadanou částku zaplatit. Hodnoty jednotlivých platidel nechť jsou sestupně zařazeny do pole – pro vytvoření pole použijte statický inicializátor.

```
int[] p = {5000, 2000, 1000, 500, 200, 100, 50, 20, 10, 5, 2, 1};
```

10. Zapište a otestujte metodu, která bude generovat pole délky n s náhodnou permutací hodnot $1..n$, kde n je celé kladné číslo, jehož hodnota nepřesáhne 1000 (realizujte dvě různé verze algoritmu / dvě různé metody).