

Úkol 1 — Sestavte funkci, která ze zadанého vektoru vypočte kumulativní součin, tj. postupný násobek hodnot v zadaném vektoru - pro vektor (a, b, c, \dots) bude výsledek $a, a \cdot b, a \cdot b \cdot c, \dots$, s číselnými hodnotami pak pro vektor 1, 2, 3, 4 bude provedeno $1, 1 \cdot 2, 1 \cdot 2 \cdot 3, 1 \cdot 2 \cdot 3 \cdot 4$, tj. výsledek je 1, 2, 6, 24.

Náhled řešení úkolu 1:

```
kumulatSoucin(1:6)  
  
## [1] 1 2 6 24 120 720  
kumulatSoucin(1)  
  
## [1] 1  
kumulatSoucin(10:13)  
  
## [1] 10 110 1320 17160
```

Úkol 2 — Tato úloha je variací na 3. úkol z cv. 3. Stáhněte si data dle úkolu 1 ve cv. 3 a vypočítejte průměrné hodnoty odtoku (Q, jednotky mm) pro jednotlivé dny v roce. Zaokrouhlete na desetiny. Vzhledem k tomu, že se jedná o denní průměry, tak použití nějakého cyklu je zde takřka nutností.

Náhled řešení úkolu 2:

```
## Kvůli velikosti proměnné denniPrumery (výsledek úlohy) je ukázána jen struktura a posledních 20 hodin  
str(denniPrumery)  
  
## Named num [1:366] 2.7 2.7 2.5 2.4 2.3 2.4 2.4 2.5 2.7 2.6 ...  
## - attr(*, "names")= chr [1:366] "01-01" "01-02" "01-03" "01-04" ...  
tail(denniPrumery, 20)  
  
## 12-12 12-13 12-14 12-15 12-16 12-17 12-18 12-19 12-20 12-21 12-22 12-23 12-24  
## 2.8 2.6 2.3 2.4 2.4 2.2 2.2 2.2 2.0 2.1 2.1 2.1 2.0  
## 12-25 12-26 12-27 12-28 12-29 12-30 12-31  
## 2.0 2.5 2.6 2.6 2.5 2.2 2.5
```

Úkol 3 — Vytvořte funkci, která zjistí v zadaném vektoru počet výskytu jednotlivých hodnot. Jedná se o variaci na úkol 1 ve cv. 5. Nepoužívejte funkci `table`, ale zkuste to jen s pomocí cyklů a funkce `unique` (a případně dalších funkcí, `unique` je zde však klíčová). Výsledek bude v podobě data.frame.

Náhled řešení úkolu 3:

```
## Příprava  
set.seed(1)  
x = sample(1:6, 20, TRUE)  
x  
  
## [1] 1 4 1 2 5 3 6 2 3 3 1 5 5 2 6 6 2 1 5 5  
## Vlastní použití vytvořené funkce  
countOccurAll(x)
```

```
## Hodnoty Pocet
## 1      1      4
## 2      2      4
## 3      3      3
## 4      4      1
## 5      5      5
## 6      6      3
```

Úkol 4 — Za pomocí for cyklu sestavte funkci, která vypíše na obrazovku n-tou hodnotu z Fibonacciho posloupnosti (n bude jediný argument funkce). Začátek posloupnosti je 0 a 1, každé další číslo posloupnosti je součtem dvou předchozích, tj. 4. prvek posloupnosti má hodnotu 2, 9. prvek pak 21 atd.

Náhled řešení úkolu 4:

```
N_fibonacci(1)
```

```
## [1] 0
```

```
N_fibonacci(10)
```

```
## [1] 34
```

```
N_fibonacci(19)
```

```
## [1] 2584
```