

Úkol 1 — Sestavte funkci, která v zadaném vektoru najde n-tou nejvyšší hodnotu.

Náhled řešení úkolu 1:

```
## Příprava
x = airquality[["Ozone"]]
str(x)

##  int [1:153] 41 36 12 18 NA 28 23 19 8 NA ...
## Vlastní použití vytvořené funkce
nthMax(x, 3)

## [1] 122
nthMax(x, 1:3)

## [1] 168 135 122
```

Úkol 2 — Sestavte funkci, která v zadaném vektoru najde celkový počet výskytu zvolené hodnoty nebo zvolených hodnot (tj. dotazované hodnoty mohou tvořit i vektor). Klíčové je zde seznámit se s operátorem %in%, protože použití známého operátoru porovnání == zde není vhodné.

Náhled řešení úkolu 2:

```
## 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
count0ccur(x, 1)

## [1] 4
count0ccur(x, 1:3)

## [1] 11
```

Úkol 3 — Sestavte funkci, která najde všechny celočíselné dělitele zadанého přirozeného čísla (kromě nuly). Předpolkádá se zde dělitelnost beze zbytku. Ošetřete funkci tak, že když na vstupu nebude správné číslo (tedy přirozené číslo větší nebo rovno 1), tak se provádění funkce zastaví s chybou. Cykly ještě nebyly představeny, tak to zkuste bez nich. Jde to.

Náhled řešení úkolu 3:

```
findDivisors(100)

## [1] 1 2 4 5 10 20 25 50 100
```

```
findDivisors(5478)
## [1] 1   2   3   6   11  22  33  66  83  166 249 498 913 1826 2739
## [16] 5478
findDivisors(6.8)
## Error in findDivisors(6.8): Zadané číslo není přirozené - je reálné
findDivisors(-5)
## Error in findDivisors(-5): Zadané číslo nevyhovuje zadání, je menší než 1
findDivisors(0)
## Error in findDivisors(0): Zadané číslo nevyhovuje zadání, je menší než 1
```