

Príklad č.1:

Vytvorte funkciu **zoznam(retazec)**, ktorá zo znakového reťazca retazec vyrobí zoznam čísel. Predpokladáme, že reťazec začína a končí znakmi '[' a ']' a prvky sú oddelené znakmi ','. Celé čísla v reťazci sa konvertujú na celé, desatinné na desatinné a nečíselné prvky sa ignorujú.

Napr.:

```
>>> z = zoznam('[0, 1., 2, 3.14]')
>>> z
[0, 1.0, 2, 3.14]
>>> zoznam('[0, -.1, None, +2, [7], a5, -3.14, "8"]')
[0, -0.1, 2, -3.14]
```

Príklad č.2:

Vytvorte funkciu **sucet(sequence)**, ktorá vypočíta „súčet“ prvkov zoznamu sequence. Predpokladáme, že všetky prvky sú rovnakého typu ako prvý prvok zoznamu. Ak je niektorý prvok iného typu ako prvý prvok, tak sa ho funkcia snaží najprv prekonvertovať na tento typ. V prípade, že sa to nedá tak takýto prvok odignoruje.

Napr.:

```
>>> sucet([2, '3', 4.0, 'päť'])
9
>>> sucet(['1', 2, 0.3, 'abc'])
'120.3abc'
>>> sucet([[1,2], 3, '4x'])
[1, 2, '4', 'x']
```

Príklad č.3:

Vytvorte funkciu `rgb(r, g, b)`, ktorá z troch celých čísel (od 0 do 255) vyrobí znakový reťazec reprezentujúci príslušnú farbu vo formáte '#rrggbb'. Funkcia pomocou troch príkazov `assert` skontroluje, či sú všetky tri parametre v poriadku.

Napr.:

```
>>> rgb(100, 150, 20.0)
...
AssertionError: chybný tretí parameter b
>>> rgb(100, 350, 20.0)
...
AssertionError: chybný druhý parameter g
>>> rgb('100', 350, 20.0)
...
AssertionError: chybný prvý parameter r
>>> rgb(100, 150, 200)
'#6496c8'
```

Príklad č.4:

V textovom súbore sa nachádza text, pričom slová sú oddelené medzerami. Súčasťou textu môžu byť aj čísla. Vytvorte funkciu `iba_cisla(meno_suboru)`, ktorá vráti zoznam celých čísel z tohto súboru. Ak sa súbor nedá otvoriť, funkcia vráti prázdny zoznam.

Napríklad, ak súbor obsahuje text: "Len 15 z uvedených 127 čísel je kladných. Najmensim je -15 a najvacsim je 4,5. Najcastejsie opakujucim sa cislom je 2."

```
>>> print(iba_cisla('subor.txt'))
[15, 127, -15, 2]
```

Príklad č.5:

Vytvorte funkcie `max(tab)`, `min(tab)` a `sum(tab)`, ktoré zistia (vrátia pomocou `return`) najväčší prvok, najmenší prvok a súčet všetkých prvkov dvojrozmernej tabuľky `tab` (bez prázdnych riadkov).

Príklad č.6:

Vytvorte funkciu `preklop(tab)`, ktorá vyrobí novú dvojrozmernú tabuľku, v ktorej bude pôvodná tabuľka preklopená okolo hlavnej uhlopriečky (vymenené riadky a stĺpce). Predpokladáme, že všetky riadky majú rovnakú dĺžku. Z matematického pohľadu funkcia `preklop` vytvára transponovanú maticu k pôvodnej matici.

Napr.:

```
>>> p = [[1, 2], [5, 6], [3, 4]]
>>> vypis(preklop(p))
  1  5  3
  2  6  4
>>> vypis(p)
  1  2
  5  6
  3  4
```

Príklad č.7:

Vytvorte funkciu `pascalov_trojuholnik(n)`, ktorá vygeneruje prvých `n` riadkov pascalovho trojuholníka a uloží ich do dvojrozmernej tabuľky. Pri vytváraní nasledovného riadka tabuľky využite predchádzajúci riadok (každý prvok v ňom je súčtom dvoch odpovedajúcich susedných prvkov z predošlého riadku).

Napr.:

```
>>> pt = pascalov_trojuholnik(6)
>>> vypis(pt)
  1
  1  1
  1  2  1
  1  3  3  1
  1  4  6  4  1
  1  5 10 10  5  1
```