

# Programovanie (1) v C/C++ 2023/24

## Cvičenia 8, príklad 3

### Matica

Vašou úlohou je **do priloženej kostry** naprogramovať dvojrozmernú verziu dátovej štruktúry dynamické pole, t.j. maticu, do ktorej je možné pridávať riadky a stĺpce. Štruktúra by mala podporovať nasledujúce funkcie:

- `init(a, m, n, x)` inicializuje štruktúru `a` ako maticu s `n` riadkami a `m` stĺpcami, všetky políčka vyplní hodnotou `x`.
- `print(a)` vypíše dynamickú maticu `a` vo formáte ako v príkladoch nižšie.
- `destroy(a)` uvoľní pamäť alokovanú pre dynamickú maticu `a`.
- `set(a, i, j, x)` nastaví v dynamickej matici `a` políčko v riadku `i` a stĺpci `j` na hodnotu `x`
- `addColumn(a, x)` do dynamickej matice `a` pridá nový stĺpec vyplnený hodnotou `x`.
- `addRow(a, x)` do dynamickej matice `a` pridá nový riadok vyplnený hodnotou `x`.

V kostre je definovaná štruktúra `dynMatrix` a funkcie `init`, `print` a `main` sú už naprogramované. Stačí teda naprogramovať funkcie `set`, `addColumn`, `addRow` a `destroy`. **Hotové časti kódu nemeňte.** Navyše sú v programe aj funkcie `vytvorMaticu`, `skopirujMaticu`, `vyplnMaticu` a `zmazMaticu`, ktoré odporúčame používať aj vo funkciách, ktoré budete programovať vy.

V štruktúre `dynMatrix` máme okrem matice `a` dve sady rozmerov. V hodnotách `m` a `n` máme počet momentálne využívaných riadkov a stĺpcov. V hodnotách `maxM` a `maxN` je počet naalokovaných riadkov a stĺpcov, čo môžu byť väčšie čísla ako `m` a `n`. Ak pri volaní `addColumn` resp. `addRow` je ešte v naalokovanej matici miesto na nový riadok alebo stĺpec, stačí ho inicializovať hodnotou `x`. Ak už nie je v matici miesto, hodnota `maxM` alebo `maxN` sa zdvojnásobí, alokuje sa väčšia matica, do nej sa presunú aktuálne prvky a stará matica sa odalokuje. Funkcia `init` inicializuje `maxM` a `maxN` na `m` a `n`. Funkcia `print` vypisuje na prvý riadok najskôr rozmery `m` a `n` a potom v zátvorke `maxM` a `maxN`.

Na vstupe sú príkazy, ktoré určujú, ktorá funkcia a s akými parametrami sa má zavolať. Prvý je vždy príkaz `init` a posledný `destroy`, medzi nimi sa striedajú ostatné príkazy.

#### Príklad vstupu:

```
init 1 1 3
print
set 0 0 0
addColumn 1
addColumn 2
addRow 3
addRow 4
addColumn 5
addColumn 6
print
destroy
```

#### Príklad výstupu:

```
1x1 (1x1)
3
3x5 (4x8)
0 1 2 5 6
3 3 3 5 6
4 4 4 5 6
```