Násobení matic, varianta 234623465345234513 2tá

Ve scheme naprogramujte násobení matic libovolné (i obdélníkové) velikosti. Matice obsahují jen normální čísla, a jsou reprezentované jako seznamy seznamů po řádkových vektorech, tj. identitní matice 3×3 je reprezentovaná '((1 0 0)(0 1 0)(0 0 1)).

Snažte se o nejkratší a nejhezčí kód (ne nutně nejrychlejší nebo nejodolnější) a co nejlepší využití předdefinovaných funkcí — hodit se může (map), (zip), (list), (foldl) a případně (apply) nebo (andmap). Většina z těchto funkcí (hlavně map a zip) jde aplikovat na libovolný počet parametrů, což se dost hodí např. při transponování matic.

Pokud je vstup chybný, můžete způsobit nedefinované chování (volně přeloženo, neřešte to).

Výslednou funkci pojmenujte mult-mtx. Funkce by měla umět zpracovat více parametrů, všechny parametry jsou matice které by se měly vynásobit postupně.

Příklad

Fungovat by to mělo zhruba takhle:

```
λ> (mult-mtx '((1 2)
               (3 4)
                  (1 0)
                  (0 1))
              '((1 2 3)
                  (4 5 6))
                '((1 0 0)
                  (0 0 1)
                  (0 1 0)))
'((9 15 12) (19 33 26) (1 3 2) (4 6 5))
```

Hint

Q: Jak vyrobit funkci s variabilním množstvím parametrů?

A: V definici funkce vyrobíte něco jako pattern-match na tělo seznamu argumentů.

```
(define (print-all-args . x)
        (for-each (lambda (x)
                          (begin (print x)
                                 (newline)))
                  x))
```