КОНТРОЛНА РАБОТА № 1 ПО ФУНКЦИОНАЛНО ПРОГРАМИРАНЕ
КН, 2-ри курс, 2-ри поток (19.11.2022 г.)

Задача 1

Да се дефинира процедура **`(primes-prod x)`**, която за дадено неотрицателно цяло число x връща произведението на простите числа, по-малки или равни на $\sqrt{x}$. Да се реализира линеен итеративен процес.

Примери:

```
(primes-prod 12)    ; → 6
(primes-prod 49)    ; → 210
(primes-prod 1200) ; → 200560490130
```

Задача 2

Да се дефинира процедура **`(shuffle-merge xs ys)`**, която приема два списъка и обединява техните елементи в списък, вземайки елементи последователно от двата списъка. Ако някой от списъците се изчерпа, всички следващи елементи трябва да бъдат взети от другия.

Примери:

```
(shuffle-merge '(1) '())                ; → '(1)
(shuffle-merge '(3 4 5) '(2))           ; → '(3 2 4 5)
(shuffle-merge '(3 4 5) '(9 2))         ; → '(3 9 4 2 5)
(shuffle-merge '(3 2 8) '(5 6 1 9 11)) ; → '(3 5 2 6 8 1 9 11)
```

Задача 3

Да се дефинира функция **`(g-l-sum limit)`**, която приема естествено число - `limit` и връща точкова двойка от две естествени числа `x` и `y` - такива, че `GCD(x, y) + LCM(x, y) = limit`, където `GCD(x, y)` е най-голямото цяло число, което дели `x` и `y`, а `LCM(x, y)` е най-малкото цяло число, което се дели едновременно от `x` и от `y`.

Приемете, че решение винаги съществува. Ако има няколко такива `x` и `y`, функцията може да върне произволна комбинация.

Примери:

```
(g-l-sum 2)  ; → '(1 . 1)
```

```
(g-l-sum 14) ; → '(6 . 4)
```

Задача 4

Разглеждаме матрица `5 × 5`, състояща се от `24` нули и една единица. Индексираме редовете на матрицата с числа от `1` до `5` отгоре надолу. Индексираме колоните на матрицата с числа от `1` до `5` отляво надясно. Един ход е прилагането на **само една** от следните трансформации към матрицата:

- Размяна на два съседни реда на матрицата, т.е. редове с индекси `i` и `i + 1` (`1 ≤ i < 5`).
- Размяна на две съседни колони на матрицата, т.е. колони с индекси `j` и `j + 1` (`1 ≤ j < 5`).

Матрицата е красива, ако единствената единица се намира в средата ѝ (в клетката, която е на пресечната точка на третия ред и третата колона).

Да се дефинира функцията **`(steps-bm xss)`**, която приема матрицата `xss` от гореописания вид, представена като списък от списъци. Функцията трябва да връща минималния брой ходове, необходими, за да се направи матрицата красива.

Примери:

```
(steps-bm '((0 0 0 0 0)
            (0 0 0 0 1)
            (0 0 0 0 0)
            (0 0 0 0 0)
            (0 0 0 0 0)) ) ; → 3

(steps-bm '((0 0 0 0 0)
            (0 0 0 0 0)
            (0 1 0 0 0)
            (0 0 0 0 0)
            (0 0 0 0 0)) ) ; → 1

(steps-bm '((0 0 0 0 0)
            (0 0 0 0 0)
            (0 0 1 0 0)
            (0 0 0 0 0)
            (0 0 0 0 0)) ) ; → 0

(steps-bm '((0 0 0 0 0)
            (0 0 0 0 0)
            (0 0 0 0 0)
            (0 0 0 0 0)
            (0 0 0 0 1)) ) ; → 4
```