

Първо контролно по Функционално програмиране

спец. Информатика, Компютърни науки, 1 поток и Софтуерно инженерство

8.11.2019 г.

Вариант А

Задача 1. а) (3 т.) Да се реализира функция `product-digits`, която намира произведението от цифрите на дадено естествено число.

б) (7 т.) Нека с $\{n\}$ означим разликата на n и произведението на цифрите на n . Да се реализира функция `largest-diff`, която намира най-голямата разлика $\{m\} - \{n\}$ за $m, n \in [a; b]$, където a и b са параметри на функцията.

Пример: `(largest-diff 28 35) → 19` = $\{30\} - \{29\} = (30 - 0) - (29 - 18)$

Задача 2. (10 т.) “Метрика” наричаме функция, която приема като параметър списък от числа и връща число като резултат. Да се напише функция `max-metric`, която приема като параметри списък от метрики `m1` и списък от списъци от числа `l1` и връща метрика `m` от `m1`, за която сумата от стойностите, които `m` връща над елементите на `l1`, е максимална в сравнение с останалите метрики от `m1`.

Примери:

```
(define (prod l) (apply * l))          (define (sum l) (apply + l))  
(max-metric (list sum prod) '((0 1 2) (3 4 5) (1337 0))) → <sum>  
(max-metric (list car sum) '((1000 -1000) (29 1) (42))) → <car>
```

Задача 3. (10 т.) “Ниво на влагане” на атом в дълбок списък наричаме броя пъти, който трябва да се приложи операцията `car` за достигане до атома. Да се реализира функция `deep-repeat`, която в подаден дълбок списък заменя всеки атом на ниво на влагане n с n негови повторения.

Пример:

```
(deep-repeat '(1 (2 3) 4 (5 (6)))) → (1 (2 2 3 3) 4 (5 5 (6 6 6)))
```

Забележка: използването на всички стандартни функции в R⁵RS, както и на функциите `accumulate`, `filter`, `foldr` и `foldl` е позволено, но не е задължително.

Първо контролно по Функционално програмиране

спец. Информатика, Компютърни науки, 1 поток и Софтуерно инженерство
8.11.2019 г.

Вариант Б

Задача 1. а) (3 т.) Да се реализира функция `sum-digit-divisors`, която намира сумата на тези от положителните цифри на дадено естествено число, които са му делители.

б) (7 т.) Да се реализира функция `same-sum`, която намира броя на двойките числа (m, n) , за които $a \leq m < n \leq b$ и функцията `sum-digit-divisors` от предното подусловие връща един и същ резултат, където a и b са параметри на функцията.

Пример: `same-sum 28 35` $\rightarrow 2$ двойките са $(28, 32)$ и $(29, 34)$

Задача 2. (10 т.) “Метрика” наричаме функция, която приема като параметър списък от числа и връща число като резултат. Да се напише функция `best-metric?`, която приема като параметри списък от метрики `m1` и списък от списъци от числа `l1` и проверява дали има метрика в `m1`, която дава по-големи стойности от всички други метрики от `m1` над всеки от елементите на `l1`.

Пример:

```
(define (prod l) (apply * l))          (define (sum l) (apply + l))
(best-metric? (list sum prod) '((0 1 2) (3 -4 5) (1337 0)))  $\rightarrow$  #t
(best-metric? (list car sum) '((100 -100) (29 1) (42)))  $\rightarrow$  #f
```

Задача 3. (10 т.) “Ниво на влагане” на атом в дълбок списък наричаме броя пъти, който трябва да се приложи операцията `car` за достигане до атома. Да се реализира функция `deep-delete`, която в даден дълбок списък изтрива всички числови атоми, които са по-малки от нивото им на влагане.

Пример:

```
(deep-delete '(1 (2 (2 4) 1) 0 (3 (1))))  $\rightarrow$  (1 (2 (4)) (3 ()))
```

Забележка: използването на всички стандартни функции в R⁵RS, както и на функциите `accumulate`, `filter`, `foldr` и `foldl` е позволено, но не е задължително.

Първо контролно по Функционално програмиране

спец. Информатика, Компютърни науки, 1 поток и Софтуерно инженерство

8.11.2019 г.

Вариант В

Задача 1. а) (3 т.) Да се реализира функция `sum-common-divisors`, която намира сумата от общите делители на две естествени числа.

б) (7 т.) Да се реализира функция `greatest-sum`, която намира най-голямата сума на общи делители на две различни числа в интервала `[a; b]`, където **a** и **b** са параметри на функцията.

Пример: `greatestSum 21 34` → 15 (за числата 24 и 32)

Задача 2. (10 т.) “Метрика” наричаме функция, която приема като параметър списък от числа и връща число като резултат. Да се напише функция `count-metrics`, която приема като параметри списък от метрики `m1` и списък от списъци от числа `l1` и връща броя на метриките от `m1`, които дават еднакви стойности за всички елементи на `l1`.

Пример:

```
(define (prod l) (apply * l))          (define (sum l) (apply + l))  
(count-metrics (list sum prod) '((0 1 2) (3 0 5) (1337 0))) → 1  
(count-metrics (list car sum) '((42 -2 2) (42 0) (42))) → 2
```

Задача 3. (10 т.) “Ниво на влагане” на атом в дълбок списък наричаме броя пъти, който трябва да се приложи операцията `car` за достигане до атома. Да се реализира функция `level-flatten`, която по подаден дълбок списък `d1` връща плосък списък от всички атоми в `d1`, като всеки числов атом е увеличен с номера на нивото си на влагане.

Пример:

```
(level-flatten '(1 (2 3) 4 (5 (6)) (7))) → (2 4 5 5 7 9 9)
```

Забележка: използването на всички стандартни функции в R^5RS , както и на функциите `accumulate`, `filter`, `foldr` и `foldl` е позволено, но не е задължително.