

Първо контролно по Функционално програмиране спец. Компютърни науки, 23.11.2011 г.

Вариант А

Задача 1. Съвършено число е естествено число, което е равно на сбора на собствените си делители. Например 6 е съвършено, тъй като 1, 2 и 3 са собствените делители на 6 и $6 = 1 + 2 + 3$.

а) Напишете функция (`perfect? n`), която връща "истина" точно когато числото `n` е съвършено.

б) Напишете функция (`sumOfPerfects a b`), която намира сумата на съвършените числа в интервала `[a, b]`. Можете да използвате функцията (`accumulate a b op null next pred? term`).

Задача 2. Напишете функция (`suffix l1 l2`), която връща "истина" точно когато списъкът `l1` е подсписък на `l2` и `l1` е в края на `l2`.

Пример: (`suffix '(1 3 5) '(2 4 1 3 5)`) \rightarrow `#t`, (`suffix '(1 2) '(4 5 1 2 7)`) \rightarrow `#f`

Задача 3. Дефинирайте функция (`consecutiveDigits? n`), която проверява дали цифрите на дадено десетично число са поредни в нарастващ ред от първата към последната.

Пример: (`consecutiveDigits? 234`) \rightarrow `#t`, (`consecutiveDigits? 531`) \rightarrow `#f`

Обяснете какъв тип изчислителен процес генерира дефинираната от вас функция.

Задача 4. Числото `x` наричаме неподвижна точка на `f`, ако $f(x) = x$. Да се напише функция от по-висок ред (`fixedPoints f a b`), която връща списък от всички неподвижни точки на функцията `f` в интервала `[a, b]` (предполагаме, че интервалът `[a, b]` се обхожда със стъпка 1).

Пример: (`fixedPoints (lambda (x) (* x x)) 0 5`) \rightarrow `(0 1)`, (`fixedPoints (lambda (x) (+ x 1)) 0 100`) \rightarrow `()`

Задача 5. Да се напише функция (`lowerThanGeometric l`), която по списък от числа `l` намира тези от елементите на `l`, които са по-малки от средното геометрично на всички числа в списъка.

Използването на `accumulate`, `map` и `filter` е позволено, но не е задължително.

Първо контролно по Функционално програмиране спец. Компютърни науки, 23.11.2011 г.

Вариант Б

Задача 1. Интересен номер е естествено число, което се дели на сбора от цифрите си. Например, 410 е интересен, тъй като $4 + 1 + 0 = 5$ е делител на 410.

- а) Напишете функция (`interesting? n`), която връща "истина" точно когато числото `n` е интересно.
б) Напишете функция (`productOfInteresting a b`), която намира произведението на всички интересни числа в интервала `[a, b]`. Можете да използвате функцията (`accumulate a b op null next pred? term`).

Задача 2. Напишете функция (`prefix l1 l2`), която връща "истина" точно когато списъкът `l1` е подсписък на `l2` и `l1` е в началото на `l2`.

Пример: (`prefix '(2 4 1) '(2 4 1 3 5)`) \rightarrow `#t`, (`prefix '(5 1) '(4 5 1 2 7)`) \rightarrow `#f`

Задача 3. Дефинирайте функция (`decreasingDigits? n`), която проверява дали цифрите на дадено десетично число са наредени в намаляващ ред от първата към последната.

Пример: (`decreasingDigits? 951`) \rightarrow `#t`, (`decreasingDigits? 958`) \rightarrow `#f`

Обяснете какъв тип изчислителен процес генерира дефинираната от вас функция.

Задача 4. Числото `x` наричаме огледална точка на `f`, ако $f(x) = f(-x)$. Да се напише функция от по-висок ред (`mirrorPoints f a`), която намира всички огледални точки на `f` в интервала `[0, a]` (предполагаме, че интервалът `[0, a]` се обхожда със стъпка 1).

Пример: (`mirrorPoints (lambda (x) (* x x)) 5`) \rightarrow `(0 1 2 3 4 5)`, (`mirrorPoints (lambda (x) (+ x 1)) 5`) \rightarrow `(0)`

Задача 5. Да се напише функция (`betterThanAverage l`), която по списък от числа `l` намира тези от елементите на `l`, които са по-големи от средното аритметично на всички числа в списъка.

Използването на `accumulate`, `map` и `filter` е позволено, но не е задължително.

Задача 1. Комплексно число е израз от вида $a + bi$, където a и b са реални числа, а i е имажинерната единица, за която е изпълнено $i^2 = -1$. Числото a наричаме реална част на комплексното число, а b - имажинерна част на комплексното число.

Като изберете подходящо представяне на *комплексно число*, напишете следните функции за работа с комплексни числа:

(complex a b) - конструктор, който създава комплексното число $a + bi$;

(re z) - селектор за реалната част на комплексното число z ;

(im z) - селектор за имажинерната част на комплексното число z ;

(add z1 z2) - намира сумата на двете комплексни числа;

(multiply z1 z2) - намира произведението на двете комплексни числа;

(abs z) - намира абсолютната стойност на комплексното число z .

Напишете кратка програма, в която демонстрирате използването на дефинираните функции.

Трифон: не съм упражнявал абстракции със структури от данни, не ми остана време, затова предлагам горната задача да отпадне

Задача 2. Напишете функция (unique l), която връща списък, съдържащ уникалните елементи на списъка l и само тях.

Пример: (unique '(2 5 "water" 5 "alpha" (5 6) "water")) връща (2 "alpha" (5 6)).

Трифон: правил съм тази задача на упражнения под името remove-duplicates

Задача 3Б

Напишете функция (duplicates l), която връща списък, съдържащ повтарящите се елементи на списъка l и само тях.

Пример: (duplicates '(2 5 "water" 5 "alpha" (5 6) "water")) връща (5 "water").

Вариант В

Задача 1.

- а) Напишете функция (exact-power? n k), която връща "истина" точно когато числото n е точна степен на числото k.
- б) Напишете функция (sum-inexact a b k), която намира сумата на x^k за всички x в интервала [a, b], които не са точна степен на числото k. Можете да използвате функцията (accumulate a b op null next pred? term).

Задача 2. Напишете функция (infix l1 l2), която връща "истина" точно когато списъкът l1 е подсписък на l2, но l1 не е нито начало, нито край на l2.

Пример: (infix '(1 3 5) '(2 4 1 3 5)) → #f, (infix '(1 2) '(4 5 1 2 7)) → #t

Задача 3. Дефинирайте функция (repeatingDigits? n), която проверява дали в дадено естествено число всяка цифра се среща поне два пъти.

Пример: (repeatingDigits? 23523) → #f, (repeatingDigits? 15521215) → #t

Обяснете какъв тип изчислителен процес генерира дефинираната от вас функция.

Задача 4. Функцията f наричаме периодична в интервала [a;b], ако има цяло число $T < b - a$, наречено период, така че $f(x+T) = f(x)$ за всяко x в интервала [a;b]. Да се напише функция от по-висок ред (period f a b), която намира най-малкия период на f в интервала [a;b]. Ако f не е периодична, функцията да връща #f.

Пример: (period (lambda (x) (remainder x 5)) 10 20) → 5, (period (lambda (x) (quotient x 5)) 10 20) → #f

Задача 5. Да се напише функция (sum-squares l), която намират сумата на всички числа от списъка l, които са точен квадрат на някое число в същия списък. Използването на accumulate, map и filter е позволено, но не е задължително.

Пример: (sum-squares '(1 2 3 4 5 25)) → 8 = 1 + 2 + 5