

**Домашна работа № 2 по Функционално програмиране**  
**специалност „Компютърни науки“, II курс, II поток**  
**2022/2023 учебна година**

---

Решенията трябва да са готови за автоматично тестване. Важно е програмният код да бъде добре форматиран и да съдържа коментари на ключовите места. Предайте решенията на всички задачи в *един* файл с наименование *hw2\_<FN>.rkt*, където *<FN>* е Вашият факултетен номер.

Домашните работи се предават като изпълнение на съответното задание в курса по ФП в Moodle (<https://learn.fmi.uni-sofia.bg/course/view.php?id=8504>) най-късно до **23:55 ч. на 23.11.2022 г.** (сряда).

*Приятна работа и успех!*

---

**Задача 1.** Да се дефинира процедура (**pair-compose fs xs**), която получава списък  $fs = (f_1 \ f_2 \ \dots \ f_n)$  от двуаргументни числови функции и списък от числа  $xs = (x_1 \ x_2 \ \dots \ x_n)$  и връща нова едноаргументна числова функция  $g$  - такава, че оценката на  $(g \ y)$  е равна на сумата  $((f_1 \ x_1) \cdot (f_2 \ x_2) \ y) + ((f_3 \ x_3) \cdot (f_4 \ x_4) \ y) + \dots + ((f_{n-1} \ x_{n-1}) \cdot (f_n \ x_n) \ y)$ , където “.” означава композиция на функции. Ако оригиналният списък с функции има нечетен брой елементи, то последната функция от списъка се композира с функцията идентитет *id*, която получава един аргумент и го връща без промяна.

*Пример:*

```
(define fs (list *
                (λ (x y) (* x x x y))
                (λ (x y) (+ x 1 y))
                (λ (x y) (- x (+ 1 y)))
                (λ (x y) (* x y 2))))
(define xs '(1 2 3 4 5))

((pair-compose fs xs) 5) → 92
; ((* 1).(* 8) 5) + ((+ 4).(- 3) 5) + ((* 10).id 5) = 40 + 2 + 50 = 92
```

**Задача 2.** Разглеждаме ситуация на санитарна сеч на гора.  $n$  дървета са разположени в точки с координати  $x_1, x_2, \dots, x_n$ . Всяко дърво  $i$  има височина  $h_i$ . Дърво може да се отсече и да се повали наляво или надясно. След това, дървото  $i$  заема един от затворените интервали  $[x_i - h_i, x_i]$  или  $[x_i, x_i + h_i]$ . Неотсеченото дърво заема една точка с координата  $x_i$ . Дърво може да се отсече, ако интервалът, който трябва да бъде зает от падналото дърво, не съдържа нито една заета точка. Дървосекачите имат задача да отсекаят възможно най-много дървета.

Да се дефинира процедура (**woodcutters xs**), която приема списък от точкови двойки от вида  $(x_i \ . \ h_i)$ , съдържащи координатата и височината на  $i$ -тото дърво, и връща максималния брой дървета, които могат да бъдат отсечени по зададените правила. Може да приемете, че точковите двойките са дадени в реда на нарастване на  $x_i$ , както и че в точка с една и съща координата няма да има две дървета.

*Примери:*

```
(woodcutters '((1 . 2) (2 . 1) (5 . 10) (10 . 9) (19 . 1))) ; → 3
```

```
(woodcutters '((1 . 2) (2 . 1) (5 . 10) (10 . 9) (20 . 1))) ; → 4
```

```
(woodcutters '((10 . 4) (15 . 1) (19 . 3) (20 . 1))) ; → 4
```

```
(woodcutters '((1 . 7) (3 . 11) (6 . 12) (7 . 6) (8 . 5) (9 . 11)
(15 . 3) (16 . 10) (22 . 2) (23 . 3) (25 . 7) (27 . 3) (34 . 5)
(35 . 10) (37 . 3) (39 . 4) (40 . 5) (41 . 1) (44 . 1) (47 . 7)
(48 . 11) (50 . 6) (52 . 5) (57 . 2) (58 . 7) (60 . 4) (62 . 1)
(67 . 3) (68 . 12) (69 . 8) (70 . 1) (71 . 5) (72 . 5)
(73 . 6) (74 . 4) ) ) ; → 10
```