

Задача 3. Задачата да се реши на един от езиките *Scheme* или *Haskell*. По-долу оградете името на езика, който сте избрали за вашето решение.

Големият онлайн магазин *Siberia* търси начин да увеличи продажбите като препоръчва на клиентите си подходящи продукти. За целта изследователският екип на *Siberia* експериментира с различни реализации на функция `bestFit`, която приема като параметър код на продукт `a` и връща код на друг продукт `b`, който клиентите на магазина най-вероятно биха си купили заедно с `a`. Задачата пред разработчиците на *Siberia* е да реализират функция `recommended`, която получава като параметри потребителска кошница `basket` (списък от целочислени кодове на продукти), функция `bestFit` и списък от продуктите на магазина `products` (списък от наредени двойки от уникален код на продукт и цена — неотрицателно число).

Да се попълнят по подходящ начин празните полета така, че функцията `recommended` да връща списък от кодовете на всички възможни препоръчани продукти. Допуска се в резултата някои кодове да се срещат повече от веднъж. Препоръчан продукт е такъв, който:

- все още не е в `basket`, но се получава като резултат от прилагането на функцията `bestFit` над някой от продуктите, които вече са в `basket`;
- има цена, която не надвишава общата цена на потребителската кошница, дефинирана като сумата от цените на продуктите в `basket`.

Помощните дефиниции `findPrice` и `basketCost` намират съответно цената на даден продукт `product` в списъка `products` и цената на потребителската кошница. Да се приеме, че `basket` съдържа само кодове на продукти в `products` и `bestFit` също връща само такива кодове.

**Упътване:** могат да се използват наготово функциите `apply`, `assoc`, `elem`, `filter`, `foldr`, `lookup`, `map`, `member`, `sum` и стандартните функции в *R<sup>5</sup>RS* за *Scheme* и в *Prelude* за *Haskell*.

#### Scheme

```
(define (recommended basket bestFit products)
  (define (findPrice product)
    _____ products _____)
  (define basketCost
    _____ basket _____)
  (
    _____
    (lambda (product)
      _____)
    (
      _____ basket)))
```

#### Haskell

```
recommended basket bestFit products =
  _____
  (\product ->
    _____)
  (
    _____ basket)

where findPrice product =
  _____ products _____
      basketCost =
  _____ basket _____
```