

ПРИМЕРЕН ТЕСТ №1 ПО ФУНКЦИОНАЛНО ПРОГРАМИРАНЕ

1. Дефинирайте функция на езика Racket, която генерира итеративен изчислителен процес. Обяснете каква задача се решава с помощта на тази функция.
2. Проследете процеса на оценяване и напишете оценката на всеки от следващите изрази на езика Racket:

```
(cdr (cadr '((a (b)) ((c (d)) e))))  
(cons '(a b) (list 'c '((d) e)))  
(append '(a (b c)) (caddr '((a b) c ((d) e))))  
(list '(a b) (list '(c d)))
```
3. Попълнете липсващия израз в дефиницията на функцията *scons* при условие, че тя получава като аргументи S-израз *a* и списък от списъци *ll* и връща като резултат списък, получен чрез добавяне на *a* в началото на всеки от елементите на списъка *ll*:

```
(define (scons a ll) (map ..... ll))
```
4. Проследете процеса на оценяване и напишете оценката на следващия израз на езика Racket:

```
(map length  
  (map (lambda (x)  
        (cond [(not (pair? x)) (list x)]  
              [(null? (cdr x)) x]  
              [else (cdr x)]))  
    '((2 3 4) (8 5) 6 (7 1 -1 5) (1))))
```
5. Дефинирайте понятието „специална форма“ в езика Racket. Посочете поне три специални форми и аргументирайте твърдението си за всяка от тях.
6. Каква е областта на действие на локалните имена, дефинирани с помощта на специалната форма **letrec** в езика Racket?
7. Обяснете действието на вградената функция за проверка на равенство **equal?** в езика Racket. Дайте поне два примера.
8. Как се извършва от интерпретаторите на Racket прилагането на дадена съставна процедура към съответните аргументи съгласно модела на средите? Дайте пример.