

ИЗПИТ ПО ФУНКЦИОНАЛНО ПРОГРАМИРАНЕ
септемврийска сесия, 2004/05 учебна година

име фак. № група

1. Какъв ще бъде резултатът от изпълнението на следната програма на езика Scheme:

```
(define (f x y)
  (if (= x 1) (- x) y))
(define (g) ((lambda () 1)))
(define (h) (h))
(f (g) (h))
```

ако реализацията на интерпретатора се основава на нормалния модел на оценяване?

- а) съобщение за грешка
б) изпълнението няма да завърши
в) 1
г) -1
2. Какъв тип изчислителен процес генерира следната програма на езика Scheme:

```
(define (f x n)
  (define (square x) (* x x))
  (cond ((= n 0) 1)
        ((even? n) (square (f x (/ n 2))))
        (else (* x (f x (- n 1))))))
```

- а) линеен рекурсивен процес
б) линеен итеративен процес
в) логаритмичен рекурсивен процес
г) логаритмичен итеративен процес
3. Напишете оценката на всеки от следващите изрази на езика Scheme:

```
(car (cdr '(a (b c) d))) → .....
(cdr (car '((a b) (c d) (e f)))) → .....
(cdr (cdr '((a b) ((c d)))))) → .....
```

4. Попълнете липсващите изрази в дефиницията на функцията merge при условие, че тя слива два списъка l1 и l2 от числа, които са сортирани във възходящ ред (резултатът също е сортиран във възходящ ред и съвпадащите елементи на l1 и l2 участват в него в толкова екземпляра, колкото пъти се срещат сумарно в l1 и l2):

```
(define (merge l1 l2)
  (cond ((null? l1) l2)
        ((null? l2) l1)
        ((<= (car l1) (car l2)) ..... )
        (else ..... )))
```

5. Какво е предназначението на следната функция на езика Scheme:

```
(define (f l1 l2)
  (cond ((null? l2) l1)
        ((member (car l2) l1) (f l1 (cdr l2)))
        (else (cons (car l2) (f l1 (cdr l2))))))
```

- а) намира сечението $l1 \cap l2$
б) намира обединението $l1 \cup l2$
в) намира разликата $l1 \setminus l2$
г) намира разликата $l2 \setminus l1$
6. Попълнете липсващия израз в дефиницията на функцията sconс при условие, че тя получава като аргументи S-израз а и списък от списъци l1 и връща като резултат списък, получен чрез добавяне на а в началото на всеки от елементите на списъка l1:

```
(define (scons a l1)
  (map ..... l1))
```

7. Нека са дадени (т.е. оценени от интерпретатора на Scheme) следните изрази:

```
(define make-mystery
  (lambda (the-list)
    (lambda (x)
      (cond ((null? x) the-list)
            ((atom? x) (set! the-list (cons x the-list)))
            (else (set! the-list (append x the-list)))
            the-list))))
(define mystery (make-mystery '(a)))
```

Напишете оценките на следващите изрази при условие, че тези изрази се оценяват точно в реда (1), (2), (3):

- (1) (mystery '()) →
(2) (mystery 'b) →
(3) (mystery '(c d)) →

8. Дефинирайте понятието “специална форма” в езика Scheme.
9. Обяснете същността на модела на заместването при оценяване на обръщения към съставни процедури в езика Scheme.
10. Каква е областта на действие на локалните имена, дефинирани с помощта на специалната форма **letrec**?
11. Дефинирайте понятието “списък” в езика Scheme.
12. Обяснете действието на вградената функция за проверка на равенство **equal?** в езика Scheme.
13. Обяснете действието на вградената функция **map** в езика Scheme. Дайте примери.
14. Как се извършва прилагането на дадена съставна процедура към съответните аргументи съгласно модела на средите?
15. Обяснете действието на вградената функция **set-cdr!** в езика Scheme. Дайте примери.
16. Анализирайте предназначението и обяснете действието на специалната форма **cons-stream** в езика Scheme.