

**ИЗПИТ ПО ФУНКЦИОНАЛНО ПРОГРАМИРАНЕ**  
**януарска сесия, 2004/05 учебна година (вариант 1)**

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 апликативния 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 a b)
  (define (f1 a b p)
    (cond ((= b 0) p)
          ((even? b) (f1 (* a 2) (/ b 2) p))
          (else (f1 a (- b 1) (+ p a)))))
  (f1 a b 0))
```

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

```
(cons 'a (cons 'b (cons 'c nil))) → .....
(append (car '((a b) (c d))) (cdr '(x y z))) → .....
(list '(a b) (list 'c)) → .....
```

4. Нека е дадена (т.е. оценена от интерпретатора на Scheme) следната дефиниция:

```
(define (f l)
  (cond ((null? l) '())
        ((atom? (car l)) (cons (car l) (f (cdr l))))
        (else (append (f (car l)) (f (cdr l))))))
```

Напишете оценката на следния израз:

```
(f '((2 (3 (4))) 5 ((6 (7)) 8))) → .....
```

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

```
(define (f l1 l2)
  (cond ((null? l2) '())
        ((member (car l2) l1) (cons (car l2) (f l1 (cdr l2))))
        (else (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. Нека е дадено (т.е. оценено от интерпретатора на Scheme) следното множество от дефиниции:

```
(define (function l)
  (if (null? l)
      '()
      (cons (car l) (function (delete-all (car l) (cdr l))))))
(define (delete-all s l)
  (cond ((null? l) '())
        ((equal? (car l) s) (delete-all s (cdr l)))
        (else (cons (car l) (delete-all s (cdr l))))))
```

Напишете оценката на следния израз:

```
(function '(a s d f c a f e d a)) → .....
```

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

```
(define make-mystery
  (lambda (the-count)
    (lambda (x)
      (cond ((null? x) the-count)
            ((number? x) (set! the-count (+ the-count x)) the-count)
            (else (set! the-count (- the-count 1)) the-count))))))
(define mystery (make-mystery 0))
```

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

(1) `(mystery '())` → .....

(2) `(mystery (+ 1 1))` → .....

(3) `(mystery 'a)` → .....

8. Обяснете същността на модела на заместването при оценяване на обръщания към съставни процедури в езика Scheme.

9. Каква е областта на действие на локалните имена, дефинирани с помощта на специалната форма `let*`?

10. Дефинирайте понятието “списък” в езика Scheme.

11. Обяснете действието на вградената функция за проверка на равенство `eq?` в езика Scheme.

12. Обяснете действието на вградената функция `apply` в езика Scheme. Дайте примери.

13. Как се извършва прилагането на дадена съставна процедура към съответните аргументи съгласно модела на средите?

14. Обяснете действието на вградената функция `set-car!` в езика Scheme. Дайте примери.

15. Анализирайте предназначението и обяснете действието на специалната форма `delay` в езика Scheme.

16. Как се дефинират процедури с променлив брой параметри? Дайте пример.