

# プログラミング言語論・テスト問題用紙

(’99年 7月 23日・ 10:30 ~ 12:00)

## 解答上、その他の注意事項

- I. 問題は、問 I ~ VII までである。
- II. 解答用紙の右上の欄に学籍番号・名前を記入すること。
- III. 解答欄を間違えないよう注意すること。
- IV. 選択式でない問で解答欄がマス目になっている場合は、1字に1マスを用いること。特に空白にも必ず1マスを用いること
- V. 解答中の文字(特に a と d)がはっきりと区別できるよう注意すること。
- VI. ノート・プリント・参考書などは持ち込み可である。
- VII. テストの配点は60点である。合格はレポートの得点を加算して、100点満点中60点以上とする。



I. 次の文章のは、プログラミング言語に関する記述である。空欄を埋めよ。答は下の選択肢から選べ。

Scheme は、分類上  (1) というプログラミング言語の一族に分類され、 (2) という言語の一方言である。関数を別の関数の引数や戻り値として自由に使うことができる。

- (1) (A) 機械語 (B) アセンブリ言語 (C) 手続き型言語 (D) オブジェクト指向言語  
(E) 関数型言語 (F) 論理型言語  
(2) (A) Algol (B) C (C) Lisp (D) Fortran (E) Java

II. 以下の通常の数学の記法を Scheme の記法にせよ。

- (1)  $9 + 7 \times 5$   
(2)  $\sin(3x + 4y)$   
(3)  $\frac{1}{\sqrt{5}-1}$  (ヒント — 平方根を求める関数は Scheme では sqrt)

III. 以下の文章の空欄をうめよ。答は下の選択肢から選べ。

「xs は ((19 99) (7 23) (10 30)) というリスト (リストのリスト) であるとする。

> (define xs '((19 99) (7 23) (10 30)))

(cdr xs) は  (1) 、 (cdr (cdr xs)) は  (2) である。(car (cdr (car (cdr xs)))) は  (3) である。」

- (1), (2) の選択肢  
(A). ((19 99) (7 23) (10 30)) (B). ((7 23) (10 30)) (C). ((10 30))  
(D). ((19 99) (7 23)) (E). ((19 99)) (F). ()
- (3) の選択肢  
(A). 19 (B). 99 (C). 7 (D). 23 (E). 10 (F). 30

IV. nil は空リスト、x は 1、y は 2、z は 3、w は 4 であるとする。

```
> (define nil '())
nil
> (define x 1)
x
> (define y 2)
y
> (define z 3)
z
> (define w 4)
w
```

リストを、cons, nil, x, y, z, w, (, ) と空白 のみ を用いて作る。たとえば (cons x (cons y nil)) は (1 2) というリストになる。

次の文章の空欄を埋めよ。

「(1) というリストは  (1) という式を、(1 3 4) というリストは  (2) という式を、((1 2) (3 4)) というリストは  (2) という式をを評価して得られる。」

ただし、(1) と (2) は次の選択肢から選べ。

- (1) の選択肢  
(A). (x)    (B). (x nil)    (C). (cons x)    (D). (cons x nil)
- (2) の選択肢  
(E). (cons x z w)  
(F). (cons x (cons z (cons w (cons nil))))  
(G). (cons x (cons z (cons w nil)))  
(H). (cons (x (cons (z (cons (w))))))

V. 次のように関数 `sum` を定義する。

```
(define (sum xs)
  (if (null? xs)                ; 式 1
      0                          ; 式 2
      (+ (car xs) (sum (cdr xs))) ; 式 3
  )
)
```

例えば、`(sum '(1 2 3))` は次のように計算される。

```
(sum '(1 2 3))
⇒ (+ 1 (sum '(2 3)))           ; 式 1が #f なので式 3が選択される
⇒ (+ 1 (+ 2 (sum '(3))))      ; 式 1が #f なので式 3が選択される
⇒ (+ 1 (+ 2 (+ 3 (sum '())))) ; 式 1が #f なので式 3が選択される
⇒ (+ 1 (+ 2 (+ 3 0)))         ; 式 1が #t なので式 2が選択される
⇒ 6
```

この例にならって、次のように定義された関数 `nth`

```
(define (nth xs n)
  (if (= n 0)                ; 式 1
      (car xs)                ; 式 2
      (nth (cdr xs) (- n 1)) ; 式 3
  )
)
```

について、`(nth '(1 2 3 4) 2)` の評価 ( 計算 ) の様子を書き下せ。

VI. 関数 `take` を次のように定義する。`(take xs n)` は、リスト `xs` の最初の `n` 個の要素からなるリストを返す。例えば、`(take '(1 2 3 4 5) 3)` は `(1 2 3)`、`(take '(5 4 3 2 1) 1)` は `(5)` である。空欄を埋め `take` の定義を完成させよ。

```
(define (take xs n)
  (if (= n 0)
      '()
      (cons (car xs) )
  ))
```

VII. 次のような関数 `map` を定義する ( この `map` は配布プリントに定義してあったものと同一である。 )

```
(define (map f xs)
  (if (null? xs)
      '()
      (cons (f (car xs)) (map f (cdr xs))))
)
```

(1) 次の式を評価した結果を答えよ。

```
(map (lambda (n) (+ (* 2 n) 3)) '(1 3 5 7 9))
```

(2) 「`(map  '(1 2 3 4 5))` を評価すると結果は `(11 13 15 17 19)` になる。」

この文章の空欄を埋めよ。ただし `lambda` 式を使い、解答欄におさまるような長さの解を考えると。



プログラミング言語論・テスト解答用紙 ('99年 7月 23日)

学部	学籍番号		氏名	
----	------	--	----	--

I. (3×2)

(1).		(2).	
------	--	------	--

II. (3×3)

(1).	
(2).	
(3).	

III. (3×3)

(1).		(2).		(3).	
------	--	------	--	------	--

IV. (3, 3, 5)

(1).		(2).	
(3).			

V. (5)

.....

.....

.....

.....

.....

.....

VI. (5)

--

VII. (5×2)

(1).	
(2).	