University of Toronto
csc324— Principles of Programming Languages, Spring 2000

Midterm Test

Duration: 50 minutes

Aids allowed: None

Make sure that your examination booklet has 5 pages (not including this one). Write your answers in the spaces provided. Write legibly.

Surname: ____________________ **First name:** ____________________

Student #: ____________________

Tutor (circle one): Anastasia Bezerianos Diana Zaiu Peeter Piegaze Sean Thompson

1.	________	/ 8
2.	________	/ 6
3.	________	/ 7
4.	________	/ 8
5.	________	/ 6
Total	________	/ 35

Question 1 [8 marks in total]

(a) The following function should return `t` iff all elements of `list` pass the given `test`. For example, `(all 'numberp '(3 5 fred 6))` should return `nil` because `fred` is not a number.

Fill in the missing function name in the 3rd line.

```
(defun all (test list)
    (cond ( (null list) t )
          ( (                test (car list))
                  (all test (cdr list))
          )
          (t nil)
    )
)
```

```
Answer: funcall
```

(b) Write a call to function `all` that will return true iff all elements of `myList` are themselves lists, and each of them has at least two elements. Assume `myList` has been given a value.

```
(all (lambda (x) (and (listp x) (>= (length x) 2))) myList)
```

(c) Complete the following Lisp function. It should return `t` iff any *single* element of `list` passes the given test.

```
(defun any (test list)

    (cond ( (null list) nil)
          ( (funcall test (car list)) t)
          ( t (any test (cdr list)))
    )
 )
```

Question 2 [6 marks in total]

Suppose we were to represent a matrix of integers in Lisp using a list, each of whose elements represents a row of the matrix. For example, the matrix

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

would be represented by the list

```
((1 2 3 4) (5 6 7 8) (9 10 11 12))
```

Complete the following Lisp function. It should return a list representing the nth column of a given matrix. For example, `(column 3 '((1 2 3 4) (5 6 7 8) (9 10 11 12)))` should return the 3rd column of our matrix, that is, `(3 7 11)`.

You may use the built-in function `(nth n list)`, which returns the nth element of the given list. It counts from zero, so for example, `(nth 0 '(A B C))` returns `A`.

```
(defun column (col matrix)
    (cond ( ( null matrix) nil)
          ( t (cons (nth (- col 1) (car matrix))
                    (column col (cdr matrix))
              )
          )
    )
 )
```

Question 3 [7 marks in total]

(a) In the starter code for assignment 2, why does `recprinttree` call a helper function, `recprintkids`? (Note: the relevant code is reprinted below.) Circle one.

1. It could be done without the helper, but it would be less efficient.
2. `recprinttree` is given a single tree to print, but when it prints its children it needs to print a list of trees.
3. Because `recprintkids` has a collector variable for building up a list of children.
4. Because `recprinttree` can't handle the bindings described in a pattern tree, but its helper function can.

Excerpt from the starter code for assignment 2:

```
; Prints a tree sideways, with the root to the far left of all its children.
; Preconditions: "tree" is a well-formed tree (with any branching factor).
(defun printtree (tree)
    (recprinttree tree 0)
)

(defun recprinttree (tree depth)
    (cond ( (null tree) (terpri) )
          ( t    (recprintkids (reverse (kids tree)) (+ depth 3))
                 (princ (make-string depth :initial-element #\ ))
                 (princ (car tree))
                 (terpri)
          )
    )
)

(defun recprintkids (kids depth)
    (cond ( (null kids) nil )
          ( t (recprinttree (car kids) depth)
              (recprintkids (cdr kids) depth)
          )
    )
)
```

(b) What is the big-oh time complexity of a call `(recprinttree t d)`, in the worst case? Define any variables that you use.

$O(n)$, where n is the number of nodes in tree t.

(c) What table would be the result of joining all three of the following database tables:

a	b	c
1	2	3
4	5	6
7	9	8

d	e	a
9	8	7
5	9	1
6	0	1

b	a	f
1	2	44
2	1	55
7	0	3
9	0	4
0	7	5
9	7	6

RESULTING TABLE:

a	b	c	d	e	f
1	2	3	5	9	55
1	2	3	6	0	55
7	9	8	9	8	6

Question 4 [8 marks in total]

Consider the following Prolog facts and rules:

```
team(diane, nlu).
team(mel, nlu).
team(suz, nlu).
team(tom, num).
team(des, num).
team(toni, theory).
team(rich, ai).
teammates(P, Q) :- team(P, X), team(Q, X).
```

(a) Express the teammates rule in logic.

```
For all p, q [There exists x [team(p,x) & team(q,x)] => teammates(p,q)]
```

or

```
For all p, q, x [team(p,x) & team(q x) => teammates(p,q)]
```

(b) What will be Prolog's first answer to the query `teammates(des, X).`?

```
X = tom
```

(c) What will be Prolog's first answer to the query `teammates(X, tom).`?

```
X = tom
```

(d) What will be Prolog's first answer to the query `team(Tom, X).`?

```
Tom = diane
X = nlu
```

Question 5 [6 marks in total]

(a) What is white box testing? Circle the one best answer.

1. White box testing occurs when we test a piece of code based on knowing only its interface.
2. White box testing occurs when we test a piece of code based on knowing its implementation details.
3. White box testing occurs when we test a sub-unit of an entire program
4. White box testing occurs when we test an entire program as a whole.

(b) Can a language that uses dynamic scope do type-checking at compile time?

1. Yes, although the type-checking is slower to do because of dynamic scope.
2. Yes, and the type-checking is just as fast as it would be with static scope.
3. No, because the scope of some or all variables is not known at compile time.
4. No, because the lifetime of some or all variables is not known at run time.

(c) Suppose the following functions have been defined in Lisp:

```
; Provides a nicer way to express a cond.
(defun my-if (test then else)
    (cond (test (eval then))
          (t    (eval else))
    )
)

; Returns the cdr of the shorter list.
(defun attempt1 (list1 list2)
    (my-if (< (length list1) (length list2))
          '(cdr list1)
          '(cdr list2)
    )
)
```

Assuming Lisp were to use *dynamic* scope, would the expression

```
(attempt1 '(A B C D) '(X Y Z))
```

cause an error? Circle "yes" or "no". If yes, describe the error; if no, give the value of the expression.

yes, and the error is:

no, and the value is: `(Y Z)`

END OF TEST