Esercizi Lisp

1) Dare dominio e codominio per la funzione Lisp TW seguente. Che funzione è calcolata da TW?

```
(de tw (f x) (funcall f (funcall f x)))
```

2) Dare dominio e codominio per la funzione Lisp FX seguente. Che funzione è calcolata da FX?

```
(de fx (f x)
  (if (= x (funcall f x))
     x
     (fx f (funcall f x))))
```

3) Dare dominio e codominio per la funzione Lisp CMP seguente. Che funzione è calcolata da CMP?

```
(de cmp (l x)
  (if (null l)
    x
    (funcall (car l) (cmp (cdr l) x))))
```

4) Sia data la funzione Lisp PMC seguente:

```
(de pmc (l x)
  (if (null l)
    x
    (pmc (cdr l) (funcall (car l) x))))
```

Dare dominio e codominio per PMC. Che funzione è calcolata da PMC? Che relazione c'è tra CMP et PMC? Come si puo definire l'una in termini dell'altra?

5) Scrivere una funzione SCELTA in Lisp, che prenda un predicato e una lista come argomento. Deve ritornare un'altra lista che contiene solo gli elementi della lista d'ingresso per i quali il predicato è vero.
Esempi:
(scelta 'symbolp '(23 + (sin 0) * x)) deve dare (+ * x)
(scelta 'numberp '(23 + (sin 0) * x)) deve dare (23)

6) Scrivere una funzione UNIONE che realizza l'operazione unione della teoria degli insiemi. Prende due liste come argomenti e produce una lista che contiene tutti gli elementi che sono in una delle due liste d'ingresso. Se un elemento figura nelle due liste deve essere presente una sola volta nella lista-risultato. Usare la funzione APPART del corso adatta ai simboli.
Esempio: (unione '(1 x) '(y 1)) deve dare (1 x y) (o altro ordine)

7) Scrivere una funzione LSIMB che prende un simbolo come argomento. Se questo simbolo è una funzione utente definita con DE, LSIMB deve produrre una lista piatta che contiene tutti i simboli che compaiono nella definizione della funzione utente.
Esempi: (lsimb 'mcd) deve dare (x y if = rest mcd)
(lsimb 'fat) deve dare (n if = * - fat) (o altro ordine)

8) Scrivere una funzione DSIMB simile a LSIMB, ma che dà solo quei simboli che sono funzioni utente, cioè introdotte con DE. La funzione DSIMB calcola allora tutte le funzioni utente chiamate direttamente del suo argomento.

Esempi: (dsimb 'mcd) deve dare (mcd rest)
(dsimb 'quad) deve dare ()

9) Data la funzione DSIMB del punto 8 di quest'esercizio. Scrivere una funzione DDSIMB, che calcola la chiusura transitiva di DSIMB. Cioè, se le funzioni nella lista di uscita di DDSIMB chiamano altre funzioni utente, debbono essere contenute anche quelle nel risultato.

Esempi: (ddsimb 'mcd) => (mcd rest)
(ddsimb 'f+) => (d n rest mcd fc f) o altro ordine

perchè la funzione F+ chiama le funzioni N, D e F; la funzione F chiama le funzioni FC e MCD; e la funzione MCD chiama la funzione REST. Tutte queste sono funzioni utente e debbono figurare nella liste del risultato.

10) Date le funzioni INSERIRE ed LINSERIRE del corso e del esercizio precedente. Scrivere una funzione SINSERIRE che unifica queste due usando FUNCALL. Cioè prende un argomento in più, il predicato della relazione d'ordine in modo simile alla funzione SUPERF del corso.

Esempio: (SINSERIRE '< 3 '(2 3 4 5)) deve dare (2 3 3 4 5)
(SINSERIRE 'L<= '(9) '(() (3 4 5))) deve dare
(() (9) (3 4 5))

11) Date le funzioni ORDINARE e LORDINARE del corso e del esercizio
precedente. Scrivere una funzione SORDINARE che unifica queste due
prendendo la relazione d'ordine come argomento supplementare usando SINSERIRE del punto 9 .

Esempio: (SORDINARE '< '(3 2 5 4)) deve dare (2 3 4 5)
(SORDINARE '> '(3 2 5 4)) deve dare (5 4 3 2)
(SORDINARE 'L<= '((2 3 4) () (3))) deve dare
(() (3) (2 3 4))