

1) Data la funzione RAZ in Lisp:

```
(de raz (x y)
  (if (< x y)
      x
      (raz (- x y) y)))
```

Quale funzione è calcolata da RAZ ?

2) Data una funzione H in Lisp:

```
(de h (n)
  (if (= 0 n)
      0
      (- n (h (- n 1)))))
```

Quale funzione è calcolata da H ?

3) Data una funzione F91 in Lisp:

```
(de f91 (x)
  (if (> x 100)
      (- x 10)
      (f91 (f91 (+ x 11)))))
```

Quale funzione è calcolata da F91 ?

4) Date due funzioni ABA et BAB in Lisp:

```
(de aba (n)
  (if (= 0 n)
      0
      (bab (- n 1))))
```

```
(de bab (n)
  (if (= 0 n)
      1
      (aba (- n 1))))
```

Quali funzioni sono calcolate da ABA e BAB ?

5) Definire una funzione QUOZ, che calcola la divisione intera sui numeri naturali usando solo somma, differenza e confronto con zero come funzioni di base.

6) Tradurre la definizione ottenuta nel punto 1 in Lisp e calcolare degli esempi sul computer.

7) Definire una funzione LG10, che calcola il numero di cifre del suo argomento in base 10; cioè più o meno il logaritmo in base 10 del suo argomento.

8) Tradurre la definizione ottenuta nel punto 4 in Lisp e calcolare esempi sul computer.

9) Scrivere una funzione PRIM, che decide se il suo argomento è un numero primo. Per esempio calcolando un fattore quando l'argomento non è primo e dare il numero stesso quando è primo.

Provare l'esempio (PRIM 32399)

## Esercizi Lisp - 16.04.2013

Siano date le funzioni F e F+ delle lezioni sui numeri frazionari rappresentati come coppie, esempio:

```
? (f+ (f 1 2) (f 1 3))  
= (5 6)
```

Altre operazioni aritmetiche in Lisp:

1) Scrivere una funzione F- che calcola la differenza tra numeri frazionari, in modo simile alla funzione F+ del corso, esempio:

```
? (f- (f 1 2) (f 1 3))  
= (1 6)
```

2) Scrivere una funzione F\* che calcola il prodotto tra numeri frazionari, esempio:

```
? (f* (f 1 2) (f 1 3))  
= (1 6)
```

3) Scrivere una funzione F/ che calcola il quoziente tra numeri frazionari, esempio:

```
? (f/ (f 1 2) (f 1 3))  
= (3 2)
```

Predicati

4) Scrivere una funzione Lisp F=, che compara due numeri frazionari. Deve dare 0, se i due argomenti rappresentano lo stesso numero frazionario, altrimenti dà 1.

Esempio:

```
? (f= (f 5 6) (f+ (f 1 2) (f 1 3)))  
= 0
```

5) Scrivere una funzione Lisp F<, che compara due numeri frazionari. Deve dare 0, se il primo argomento è più piccolo del secondo, altrimenti dà 1. Esempio:

```
? (f< (f 1 2) (f 1 3))  
= 1
```

6) Scrivere una funzione Lisp F>, che compara due numeri frazionari. Deve dare 0, se il primo argomento è più grande del secondo, altrimenti dà 1. Esempio:

```
? (f> (f 1 2) (f 1 3))  
= 0
```

Esercizi Lisp - 17 aprile 2013

1) Data la funzione PEX in Lisp:

```
(de pex (x y)
  (cond
    ((= 0 y) 1)
    ((= 0 (rest y 2)) (quad (pex x (/ y 2))))
    (t (* x (pex x (- y 1)))))
```

Dare dominio e codominio per PEX.  
Che funzione è calcolata da PEX?

2) Scrivere una funzione LQUAD in Lisp, che prende una lista di numeri come argomento e che produce una lista con tutti questi numeri al quadrato.

Esempio: (lquad (list 3 25 13)) deve dare (9 625 169)

3) Fare come in 2 con la funzione Fattoriale al posto del quadrato.

4) Scrivere una funzione LPARI in Lisp, che prende una lista di numeri come argomento e che produce una lista che contiene sole i numeri pari della lista d'ingresso, ma non i numeri dispari. Se non ci sono numeri pari produce la lista vuota.

Esempio: (lpari (list 49 22 5 40 3)) deve dare (22 40)

5) Fare la stessa cose come in 4, ma per numeri primi invece di numeri pari: Scrivere una funzione LPRIM in Lisp, che prende una lista di numeri come argomento e produce una nuova lista come risultato che contiene solo i numeri primi della lista d'ingresso.

Usare la funzione PRIM dell'esercizio precedente.

Esempio: (lprim (ll 20)) deve dare (19 17 13 11 7 5 3 2)

6) Data la funzione LDIV in Lisp:

```
(de ldiv (x y)
  (cond
    ((null y) ())
    ((= 0 (rest (car y) x)) (ldiv x (cdr y)))
    (t (cons (car y) (ldiv x (cdr y)))))
```

Dare dominio e codominio per LDIV.  
Che funzione è calcolata da LDIV?

7) Scrivere una funzione FPRIM in Lisp che fa la scomposizione di un numero in fattori primi. Cioè la funzione prende un numero come argomento e produce una lista che contiene tutti i suoi fattori primi.

Esempio: (fprim 16762) deve dare (2 17 17 29)  
e per ognuno numero x si verifica (prodlist (fprim x)) = x

8) Data la funzione VER in Lisp :

```
(de ver (x y)
  (if (null x)
      y
      (ver (cdr x) (cons (car x) y))))
```

Dare dominio e codominio per VER.  
Che funzione è calcolata da VER?

9) Scrivere una funzione LPIU in Lisp che prende due liste come argomenti e che decide se queste liste hanno la stessa lunghezza, senza calcolarla naturalmente; cioè senza usare operazioni aritmetiche. Quando i due argomenti hanno la stessa lunghezza, la funzione LPIU deve dare 0, altrimenti 1.

Esempio: (lpiu (ll 4) (list 1 2 3 4)) deve dare 0

10) Prodotto scalare di vettori.

Un vettore può essere rappresentato da una lista. Scrivere una funzione SCALPROD in Lisp, che prende due liste come argomento, e che produce un numero che è il prodotto scalare dei argomenti considerati come vettori. Naturalmente, le due liste hanno la stessa lunghezza.

Esempio: (scalprod (list 3 2 1) (list -1 0 1))  
deve dare -2

1) Data la funzione SART2 in Lisp:

```
(de sart2 (x)
  (list
    (list (car (car x)) (car (cadr x)))
    (list (cadr (car x)) (cadr (cadr x)))))
```

Dare dominio e codominio per SART2.  
Che funzione è calcolata da SART2?

2) Scrivere una funzione DET2 in Lisp, che prende una matrice quadrata di dimensione 2 come argomento e che calcola il suo determinante. Una matrice può essere rappresentata come una lista di righe, dove una riga è una lista piatta di numeri;

```
( 1 2 )
( 3 4 ) sarà la lista ((1 2) (3 4)) .
```

Esempio: (det2 (list (list 1 2) (list 3 4))) deve dare -2

3) Data la funzione LLL in Lisp

```
(de lll (x)
  (if (null x)
    ()
    (cons (ll (car x)) (lll (cdr x)))))
```

Dare dominio e codominio per LLL.  
Che funzione è calcolata da LLL?

4) Scrivere una funzione LLONG in Lisp che prende una lista di liste come argomento e che produce una lista di lunghezze di queste sottoliste.

Esempio: (llong (list (ll 20) () (list 3))) deve dare (20 0 1)

5) Scrivere una funzione L<= in Lisp che prende due liste come argomenti e confronta le lunghezze, naturalmente senza calcolarle.  
Se il primo argomento ha una lunghezza minore o uguale a quella del secondo argomento, la funzione L<= deve dare T, altrimenti deve dare la lista vuota ().

Esempi: (L<= (ll 2) ()) deve dare ()  
(L<= (ll 2) (ll 3)) deve dare t .

6) Scrivere una funzione LINSERIRE in Lisp che prende una lista piatta e una lista di liste come argomenti. Il primo argomento è una lista qualsiasi, il secondo argomento contiene le liste ordinate sulla loro lunghezza in modo crescente.

La funzione LINSERIRE deve inserire il primo argomento al posto giusto nel secondo per mantenere la relazione d'ordine, in modo simile alla funzione INSERIRE per numeri.

Usare il predicato L<= del punto 5.

Esempio: (LINSERIRE (ll 2) (list () (ll 4) (ll 5))) deve dare  
(() (2 1) (4 3 2 1) (5 4 3 2 1))

7) Scrivere una funzione LORDINARE che prende una lista di liste come argomento e che produce una lista come risultato che contiene le stesse sottoliste dell'argomento, ma ordinata in base alla lunghezza crescente. Usare la funzione LINSERIRE del punto 6 in modo simile alla funzione ORDINARE per numeri.

Esempio: (LORDINARE (list (ll 3) () (list 2 4))) deve dare

(( ) (2 4) (3 2 1))

8) Data la funzione BLAMOS in Lisp:

```
(de blamos (x)
  (cond
    ((null x) 0)
    ((atom x) x)
    (t (+ (blamos (car x)) (blamos (cdr x))))))
```

Dare dominio e codominio per BLAMOS.  
Che funzione è calcolata da BLAMOS?

9) Scrivere una funzione PRODALB in Lisp che prende una lista qualsiasi come argomento e che calcola il prodotto di tutti gli numeri che si trovano ad un qualsiasi livello. Cioè la lista contiene normalmente anche sottoliste.

Esempio: (prodalb (list 3 (ll 3) (list -2 4))) deve dare -48 .

10) Scrivere una funzione PIATTA che prende una lista di liste come argomento e che produce una lista piatta come risultato che contiene tutte le foglie del argomento, cioè che non contiene più sottoliste.

Esempio: (piatta (list 2 (list 3 4) (list 2))) deve dare (2 3 4 2)