

Esame di Linguaggi di Programmazione AA 2018-2019 - E2S

19 febbraio 2019

Domanda 1

Data la seguente funzione LISP:

```
(defun multichan (functions &rest vals)
  (if (null functions)
      ()
      (cons (apply (first functions) value)
            (apply #'multichan (rest functions) vals))))
```

descrivetene il comportamento e calcolate il valore che si ottiene passandogli i due argomenti seguenti

```
(list #'list #'cdr (lambda (x) (+ 42 (first x) 42)) #'length)
(list 42 666)
```

Domanda 2

Costruite la funzione `reduce` (o chiamata anche `foldl`) che prende una funzione binaria e una lista e produce come risultato la combinazione (associativa a sinistra) degli elementi della list. Potete assumere che la lista abbia almeno due elementi.

Esempi:

```
(reduce #'- '(1 2 3 4)) ==> (- (- (- 1 2) 3) 4) => -8
(reduce #'* '(1 2 3 4 5)) ==> 120
(reduce #'list '(1 2 3 4)) ==> (((1 2) 3) 4)
```

Domanda 3

Scrivete la definizione in C per una “matrice”; la definizione deve essere una `struct` che contiene il numero di righe, il numero di colonne e l’array (di `doubles`) che contiene ogni elemento della matrice. Definite anche una `typedef matrix` appropriata per descrivere puntatori alle strutture così definite.

Scrivete la funzione `new_matrix` che prende il numero di righe e il numero di colonne e restituisce una nuova matrice con tutto lo spazio necessario ottenuto tramite `malloc`.

Domanda 5

Dato il seguente programma

```
appendi([], Xs, Xs).  
appendi([X | Xs], Ys, [X | Zs]) :- appendi(Xs, Ys, Zs).
```

quali sono le soluzioni delle queries qui sotto?

1. `?- appendi([1, 2], [[3, 4], 5], X).`
2. `?- appendi([vediamo, se, la], [[becco]], X).`
3. `?- appendi([vediamo], [X | Xs], [vediamo, [se, la, becco]]).`
4. `?- appendi(X, Y, [1, 2, 3]).`

Domanda 4

Si consideri il seguente programma Prolog:

```
tower(X) :- block(X).  
tower(t(X,Y)) :- block(X), tower(Y), on(X,Y).  
block(a).  
block(b).  
block(c).  
block(d).  
block(e).  
on(b,a).  
on(d,b).
```

Si mostri l'albero SLD relativo alla regola di selezione *right-most* per la query:

```
?- tower(G).
```

Domanda 6

Scrivere il programma Prolog che data una lista IN e due elementi VECCHIO e NUOVO, sostituisce ad *ogni* elemento uguale a VECCHIO nella lista IN il valore NUOVO e lo inserisce in una lista OUT.

Esempio:

```
?- sostituisci(cane, gatto, [il, cane, si, è, grattato], Out).  
OUT = [il, gatto, si, è, grattato].
```