

Esame di Linguaggi di Programmazione (E3)

18 Giugno

Domanda 1

Supponete di avere un *header file* C su cui avete il controllo: *my_stuff.h*. Il file *my_stuff.h* `#include` il file *API.h*, il quale `#include` il file *basicAPI.h*. Mostrate come devono essere scritti i files *my_stuff.h*, *API.h* ed un vostro altro file *main.c* al fine di evitare inclusioni multiple di header files. Il file *main.c* deve includere separatamente ed esplicitamente i due files *basicAPI.h* e *my_stuff.h*.
Notate che il file *basicAPI.h* si trova nelle cartelle controllate dal sistema, mentre il file *my_stuff.h* si trova nella cartella dove state sviluppando il progetto.

Domanda 2

Considerate la seguente espressione aritmetica:

$\tan(\pi * 4.2) / (12 - e * 3) + 1024$

1. Riscrivetela in forma S-expression.
2. Scrivete una funzione *value* che ritorni il valore dell'espressione rappresentata da S-espressioni di questo tipo: considerate solo le costanti *pi* ed *e*, e gli operatori *+*, *-*, */*, *sin*, *cos*, e *tan*. Se vi servono altre funzioni ausiliarie, scrivetele pure.)

Esempi:

```
prompt> (value 42)
42
prompt> (value '(+ 40 (- 4 2)))
42
prompt> (value '(+ 42 (* (sin pi) 1024)))
42.0
```

eval!

$= \left[\begin{array}{l} \text{(defun value (x))} \\ \text{(eval x)} \end{array} \right]$

Attenzione all'identazione!

Domanda 3

Una funzione fondamentale in (Common) Lisp, con varianti equivalenti in moltissimi altri linguaggi funzionali, logici e non, è la *CONS*, (in Java e C++ l'operatore *new*, etc. etc.)

1. Quale funzionalità deve fornire l'ambiente Lisp a corredo di quest'operazione fondamentale?
2. Ovvero, senza questa funzionalità, che potrebbe accadere durante l'esecuzione di questo programma?

```
(defun giro-giro-tondo (n 11 12)
  (when (plusp n)
    (append 11 12)
    (frulla-frulla (1- n) 11 12)))
```

```
prompt> (giro-giro-tondo quarantadue-fantastiliardi '(4) '(2))
```


LINGUAGGI DI PROGRAMMAZIONE - GIUGNO 2013

(Lisp + C)

ES 1 (soluzione)

```
→ my-file.c
#include "my-stuff.h"
#include <basicAPI.h>

→ my-stuff.h
#ifndef MY_STUFF_H
#define MY_STUFF_H
#include "API.h"
#endif

→ API.h
#ifndef API_H
#define API_H
#include <basicAPI.h>
#endif
```

NB1

- uso di `<>` quando il file header si trova nella directory del compilatore di sistema
- uso di `" "` quando il file header si trova nel file del compilatore locale al progetto

blocco 3 pg 22-23-24

NB2

- `const char* PC`
puntatore a char di tipo readonly
- `const char * PC`
costante di tipo char puntatore variabile
- `char* const PC`
puntatore di tipo readonly a char
- `char * const PC`
tipo char puntatore costante variabile
- `const char* const PC`
entrambi (puntatore + variabile) readonly