

Anteprima di test

parte pratica 4 Febbraio 2015

Data: Thu Feb 19 11:33:01 2015 Punteggi massimi: 31

1. Ricorsione - C - multipli di 7 (12 Punti)

Scrivi una funzione `multiplo7` in C che dato in ingresso un numero, dice se è multiplo di 7 (senza usare %). Usa la regola: Un numero è divisibile per 7 se la somma tra il numero ottenuto escludendo la cifra delle unità (prenumero) e il quintuplo della cifra delle unità (coda numerica) è 7 o un multiplo di 7. Al solito, scrivi tre versioni: una non ricorsiva, una ricorsiva senza tail recursion e una ricorsiva con tail recursion.

Specifica esattamente con commenti i parametri che passi alla procedura, il tipo di passaggio utilizzato e il loro significato.

Definisci funzioni ausiliarie di aiuto se necessario, per tenere la segnatura della funzione `multiplo7` più semplice.

Scrivi anche un main di esempio in cui chiami la funzione in tre modi diversi:

1. non ricorsiva con il numero 68089 e assegna il risultato ad una variabile M7L locale al main
2. ricorsiva non tail con numero 14 e in modo assegni il risultato ad una variabile globale M7G. Non usare alcuna altra variabile globale.
3. ricorsiva tail con numero 6853 in modo che stampi semplicemente il risultato.

Disegna il record di attivazione per tutte e tre le versioni fino alla massima estensione del record di attivazione. Nel caso di tail recursion, spiega nel codice quali ottimizzazioni hai adottato o potresti adottare.

Leggi le istruzioni qui:

Istruzioni per l'esame di info 3 – funzioni ricorsive

Uso di C

Crea un nuovo progetto C con eclipse. Metti TUTTO dentro il progetto in eclipse (anche il foglio elettronico).

Record di attivazione

Se vuoi puoi anche disegnare il record di attivazione usando openoffice (oocalc)

Usa questo formato. Ogni record disegnalo come blocco con bordi di celle così:

Etichetta	Nome variabile	valore	Indirizzo Puntato
foo(x,y) [1]	CL		
	x	6	
	y	7	
[6]	AL		[1]
blockinline	CL		
	...		

Nella prima colonna metti la descrizione del tipo di RA. Metti anche un numero o label se vuoi riferirti a quell'indirizzo

Qui metti descrizione della cella (ad esempio nome variabile, CL ...)

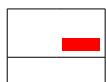
Metti il valore ospitato (se non è un indirizzo)

Metti il valore (o etichetta) se è un indirizzo

Aggiungi liberamente commenti a destra

Se devi fare vedere come evolve, copia, incolla a dex e prosegui in basso:

RA 1



Crea un progetto in eclipse con nome **COGNOME_NOME_RIC**. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome **COGNOME_NOME_RIC.zip**. Consegna il file ZIP qui:

(12 Punti)

2. Passaggio per riferimento (4 Punti)

Fa un esempio in C++ in cui scrivi una funzione che calcola il prodotto di due numeri interi mediante somme ripetute (cioè senza usare l'operazione prodotto). Passa entrambi i fattori per riferimento e usa il primo parametro anche per restituire il valore (cioè la funzione sia void).

Quali sono i limiti dell'uso di questa tecnica? Illustra con qualche esempio nel main.

Commenta nel codice cosa stai facendo.

Crea un progetto in eclipse con nome COGNOME_NOME_REF. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME_NOME_REF.zip. Consegna il file ZIP qui:

(4 Punti)

3. Java visitor pattern - Figure geometriche (5 Punti)

Una figura geometrica può essere o un quadrato o un rettangolo o un cerchio. Scrivi le classi per rappresentare le figure di questo tipo.

Si vuole realizzare un calcolatore di area e uno di perimetro di queste figure. Realizzalo in tre modi (possibilmente in tre package diversi):

1. con due metodi delle figure che restituiscano il valore dell'area e del perimetro
2. usando due classi CalcolatoreArea e CalcolatorePerimetro con un metodo per ogni classe (getArea(rettangolo)).
3. mediante il visitor design pattern

Scrivi un main in cui costruisci qualche figura e calcoli l'area e il perimetro usando l'approccio i tre approci.

Crea un progetto in eclipse con nome COGNOME_NOME_JAVA. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME_NOME_JAVA.zip. Consegna il file ZIP qui:

(5 Punti)

4. definizione di funzione in SCALA - contaMultipli7 Collezioni (6 Punti)

- **Scrivi una funzione contaPari che data una lista di interi (List[Int]) restituisce il numero di interi pari in essa. Puoi fare diverse versioni (in ordine di difficoltà):**
 - contandoli con un ciclo for semplice (for (n <- nums))
 - contandoli ma usando un foreach (nums.foreach(countPar)) e una funzione countPar
 - in modo ricorsivo (usa isEmpty head e tail)
 - usando folderLeft, del tipo (nums.foldLeft(...)(...))
 - usando filter (nums.filter ..)
- **Scrivi una higher order function contaP che generalizza il conteggio con un predicato generico P (funzione da int a boolean) che dato un intero x mi dice se x soddisfa P. contaP prede una lista di interi (List[Int]) e conta il numero di interi nella lista che soddisfano P. Usa il currying se riesci.**
- **Riscrivi (con nome contaPari2) la funzione contaPari usando contaP (sia con funzione anonima per sapere che un numero è pari sia con funzione esplicita).**
- **Scrivi un po' di chiamate definendo un Object e usando lo schema seguente:**

```
1 object prova {
2
3   def ....
4
5   def main(args: Array[String]) {
6     println( ....)
7   }
8 }
```

Crea un progetto in eclipse con nome COGNOME_NOME_SCALA. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME_NOME_SCALA.zip. Consegna il file ZIP qui:

(6 Punti)

Introduction to Functional Programming with Scala

Angelo Gargantini

INFO 3A AA 2013/14
credits: Pramode C.E
<https://class.coursera.org/progfun-00>

December 9, 2013

Workshop Plan

Here is what we will do:

- Learn a bit of functional programming in Scala
- Learn some important concepts like (NOT ALL): closures, higher order functions, purity, lazy vs strict evaluation, currying, tail calls/TCO, immutability, persistent data structures, type inference etc!

Workshop material (slide/code samples) sul sito.

5. Dangling pointer in C++ (4 Punti)

Fa un esempio in C++ in cui hai un dangling pointer sullo stack di un oggetto vero e proprio e continui ad usarlo prima senza accorgerti (cioè senza alcun malfunzionamento) e poi fai in modo invece di notare qualche malfunzionamento.

Crea un progetto in eclipse con nome COGNOME_NOME_DP. Per la consegna produci un file zip usando la funzione export archive file di eclipse. Lo zip abbia nome COGNOME_NOME_DP.zip. Consegna il file ZIP qui:

(4 Punti)