

Esercitazione 8

09 gennaio 2018

Termine per la consegna dei lavori: **martedì 16 Gennaio 2018** ore **23.59**.

Istruzioni

Ogni esercizio dovrà esser salvato in un file con estensione `.py` e consegnato tramite Moodle su sito <https://elearning.unipd.it/math>, sezione “**Matematica: Laurea Triennale**” corso di “**Laboratorio di Programmazione**” cliccando su “**Esercitazione8**”.

È obbligatorio che all’interno di ogni file sia riportato il vostro nome, cognome e numero di matricola (potete riportarli all’interno di una riga commento all’inizio del file, es: `#Mario Rossi 1234567`).

Controllate che l’esecuzione del comando:

```
python <nome_file>.py
```

per ognuno degli esercizi produca l’output desiderato.

ATTENZIONE!

Gli unici moduli importabili ammessi in questa esercitazione sono i moduli `math` e `random`. Esercizi risolti utilizzando altri moduli importati riceveranno il punteggio minimo. Python contiene delle **built-in functions**, che potete utilizzare e la cui lista si può trovare a questo indirizzo: <https://docs.python.org/2/library/functions.html>.

Esercizio 1

Un albero binario T è chiamato *albero di Fibonacci* se è un albero vuoto, oppure se per ogni nodo n di T vale la seguente proprietà: n è una foglia oppure le altezze dei sottoalberi destro e sinistro di n differiscono *esattamente* di 1.

Definire la funzione `isFibonacci(T)` che restituisca `True` o `False` a seconda che T sia o meno un albero di Fibonacci. Si utilizzi la classe `Albero` costruita in laboratorio e tutte le funzioni di utilità che possono esservi utili.

Riportare almeno due esempi di invocazione, uno su un albero di Fibonacci non vuoto e uno che ritorni `False`.

http://it.wikipedia.org/wiki/Albero_di_Fibonacci

Esercizio 2

Chiameremo albero binario natalizio, un albero binario *n-bilanciato* i cui nodi contengono le decorazioni. Un albero è *n-bilanciato*, quando, per ogni sotto-albero radicato in un suo nodo, il numero dei nodi del sotto-albero sinistro meno il numero dei nodi del sotto-albero destro è in valore assoluto al più 1.

Si chiede di creare una classe `XmasTree` (che è un albero binario) e le seguenti funzioni **ricorsive**:

- una funzione per stampare l'albero in modo formattato. Non viene richiesto nulla di artistico, si deve capire la struttura dell'albero e i valori nei nodi;
- una funzione `build_tree(decorations)` che accetta come parametro una lista di decorazioni (di qualsiasi tipo) e ritorna un albero binario natalizio con le decorazioni disposte nei suoi nodi (l'ordine di inserimento non è importante);
- una funzione `is_xmas_tree(tree)` che ritorna vero se l'albero passato come parametro è un albero binario natalizio (ovvero un albero *n-bilanciato*) altrimenti ritorna falso.

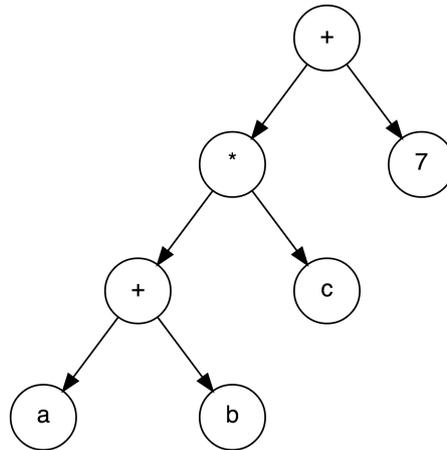
Se lo trovate necessario potete definire altre funzioni che possono aiutarvi nel risolvere i punti appena elencati. Dare almeno un esempio di utilizzo delle funzioni create.

Esercizio 3

Un'espressione matematica può essere rappresentata come un albero binario in cui i nodi rappresentano le operazioni e le foglie rappresentano gli operandi. Ad esempio l'espressione

$$(a + b)c + 7$$

può essere rappresentata dall'albero:



Scrivere una funzione `valuta (T)` che, dato un albero `T` che rappresenta una espressione, ne ritorni il suo valore. Si utilizzi la classe `Albero` costruita in laboratorio e tutte le funzioni di utilità che possono esservi utili.

Riportare almeno un esempio di invocazione su un albero che rappresenta un'espressione.