

14 Giugno 2018

Esame di Fondamenti di Informatica per Ing. Gestionale

Compito B - Durata 1h45'

Istruzioni (leggere attentamente)

Note

La mancata osservanza delle seguenti regole può comportare la perdita di informazioni necessarie alla valutazione dell'esame.

Cartella di esame e registrazione dei dati dello studente

La cartella Esame contenente il compito da svolgere si trova sul Desktop. Entrare nella cartella e, prima di iniziare il compito, eseguire il programma `registrazione.pyc`. Inserire i dati personali fornendo (separatamente) *Numero di Matricola*, *Cognome* e *Nome*. Il programma genera un file `studente.txt` che non deve essere modificato manualmente. Verificare che i dati nel file `studente.txt` siano corretti; in caso di errore potete rieseguire il programma `registrazione.pyc`.

Svolgimento degli esercizi

Leggere attentamente il testo e risolvere gli esercizi proposti.

Per ogni esercizio avete una cartella `EsercN` che contiene un file dal nome `B_ExN.py` (dove N è il numero dell'esercizio) con lo scheletro della soluzione. Il file `B_S_ExN.pyc` contiene la soluzione precompilata dell'esercizio proposta dal docente. L'unico file da modificare è `B_ExN.py`. *Nel vostro interesse, non modificate altri file (ad esempio contenenti dati di test). Non create nuovi file.*

Per verificare la correttezza di un esercizio **DOVETE** usare il programma `TestEx.pyc`, che proverà la vostra soluzione su un certo numero di casi di test, mostrandovi il confronto tra i risultati ottenuti con la vostra soluzione e usando la soluzione proposta dal docente. Ciò ha lo scopo di facilitarvi nella ricerca e correzione di eventuali errori. *Ricordiamo comunque che in sede di correzione da parte dei docenti le vostre soluzioni verranno provate su dati di test diversi da quelli usati durante l'esame*

E' possibile consultare la documentazione ufficiale del linguaggio Python (ad esempio premendo F1 all'interno dell'ambiente di programmazione Idle), ma non è possibile usare libri o appunti. In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

Esercizi

1. **B_Ex1(s) (9 punti)** Completare la funzione Python `B_Ex1(l, s)` che prende in ingresso una lista di stringhe `l` e una stringa `s`. La funzione deve restituire una lista di booleani così composta: al posto `i`-esimo, `True` se la stringa `s` compare come prefisso, *indipendentemente dal caso*. Ad esempio, "Casa" e "casa" sono la stessa stringa ai fini dell'esercizio.

Esempio: se `l` fosse: `["mama", "eluma", "MAmaci"]` e `s` fosse `Ma` la funzione dovrebbe restituire la lista `[True, False, True]`

2. **B_Ex2(m) (9 punti)** Completare la funzione Python `B_Ex2(m, min, max)` che riceve in ingresso una matrice `m` di interi rappresentata per righe come lista di liste e due interi `min` e `max`. La funzione deve restituire una lista `l` di booleani. In particolare, l'`i`-esimo elemento di `l` deve valere `True` se la somma degli elementi della `i`-esima riga di `m` è compresa fra `min` e `max`. L'`i`-esimo elemento di `l` deve valere `False` altrimenti. Se `m` è la vuota `l` deve essere la lista vuota.

Esempio 1: se la matrice fosse

$$\begin{pmatrix} 1 & 2 & 3 \\ 5 & 6 & 2 \\ 1 & 3 & -8 \\ 1 & 12 & 1 \end{pmatrix}$$

e `min` fosse `0` e `max` fosse `13`, la funzione restituirebbe `[True, True, False, False]`.

3. **B_Ex3(file) (9 punti)** Implementare la funzione `B_Ex3(file)` che prende in ingresso il nome di un file in formato csv (ossia valori alfanumerici separati da virgole, *senza spazi*) contenente le spese effettuate da una persona:

```
...  
Nome, Spesa1, Spesa2, ... , SpesaN  
...
```

La funzione deve restituire un dizionario contenente, per ogni persona, la spesa complessiva.

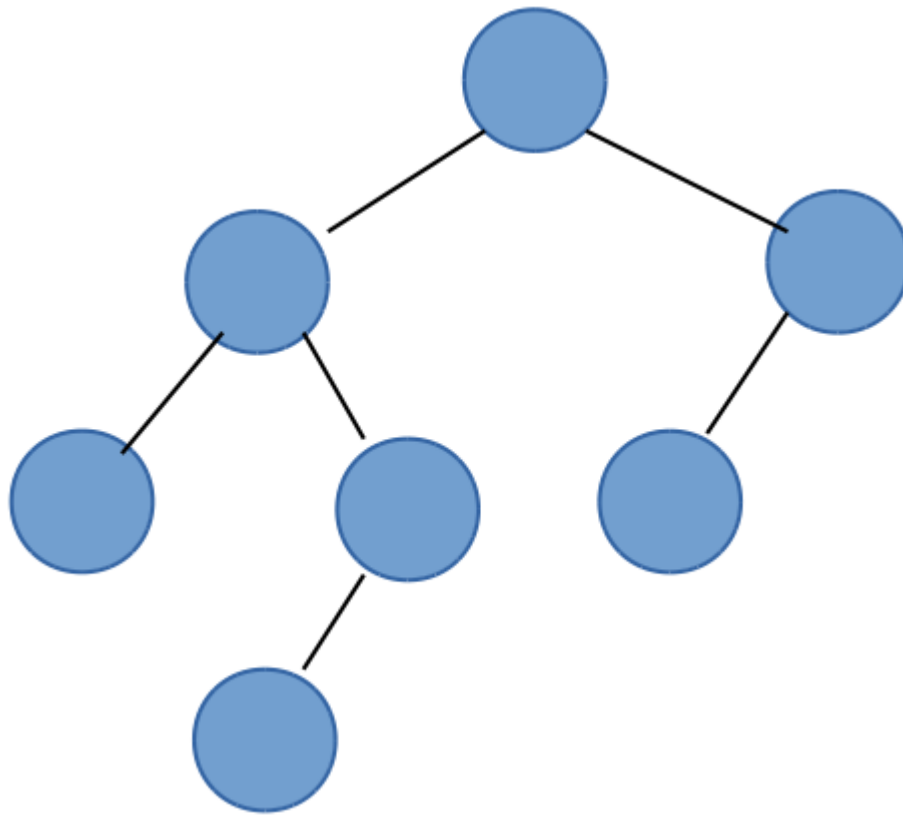
****Esempio**:** se il file fosse il seguente:

```
Francesco, 10, 20, 30  
Luca, 11, 29, 10
```

la funzione dovrebbe restituire il dizionario `{Francesco : 60, Luca : 50}`. Si assuma che ogni persona compaia una e una sola volta all'interno del file, che i valori della sua spesa siano interi e che abbia effettuato almeno una spesa.

4. **B_Ex4(a) (5 punti)** Implementare la funzione `B_Ex4(a)` che riceve in ingresso un oggetto `a` della classe `BinaryTree`, che rappresenta un albero binario e la cui interfaccia è riportata sotto. La funzione deve restituire il numero di nodi dell'albero esclusi quelli che hanno esattamente 2 figli. Se l'albero è vuoto la funzione deve restituire il valore 0.

Ad esempio, se l'albero fosse quello della figura sottostante la funzione dovrebbe restituire il valore 5.



La classe `BinaryTree` implementa, tra gli altri, i metodi descritti dalla seguente interfaccia:

```
class BinaryTree:
    ## Il generico oggetto di questa classe rappresenta la radice di un (sotto)albero
    ## binario e contiene i riferimenti agli eventuali figli destro e sinistro
    def __init__(self, rootObj):
        ## Inizializza un albero. rootObj è l'identificatore della radice

    def getRightChild(self):
        ## Restituisce la radice del sottoalbero sinistro o None (se il nodo corrente
        ## non ha un figlio sinistro)

    def getLeftChild(self):
        ## Restituisce la radice del sottoalbero sinistro o None (se il nodo corrente
        ## non ha un figlio sinistro)

    def getRootVal(self):
        return self.key
        ## Restituisce l'identificatore associato al nodo rappresentato da self

    ## Altri metodi
```

Si noti che potreste non aver bisogno di tutti i metodi (o del costruttore) elencati sopra. Si noti anche che copia dei dati di test a partire dai quali sono costruiti gli oggetti passati alla funzione si trovano nei file `file1.txt ... file8.txt`. Non modificare per nessun motivo gli originali contenuti nella sottocartella `files`