

14.09.2017

# Esame di Fondamenti Informatica per Ingegneria Gestionale - A.A. 2016/2017

## Durata 1h45' - Compito B

### Istruzioni (leggere attentamente)

**Nota importante:** 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). Questo è il file che dovrete modificare con la vostra soluzione. Non create nuovi file.

Per verificare la correttezza di un esercizio **DOVETE** usare il programma TestEx.pyc (basta cliccarci sopra 2 volte) che proverà la vostra soluzione con un certo numero di casi di test. **Si noti che per la correzione verranno usati insieme di dati di test diversi.**

E' possibile consultare la documentazione ufficiale del linguaggio Python, ma **non è possibile usare libri o appunti**. In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

#### Esercizi

- **B\_Ex1(s) (8 punti)** Completare la funzione Python **B\_Ex1(s)** che prende in ingresso una stringa s. La funzione deve restituire una stringa ottenuta da s nel seguente modo. La stringa da restituire è la concatenazione di len(s) stringhe, la i-esima delle quali è ottenuta ripetendo l'i-esimo carattere di s (i+1) volte (quindi 1 volta il carattere in posizione 0, due volte il carattere in posizione 1 e così via). Qualora s sia la stringa nulla la funzione deve restituire la stessa stringa.

**Esempio:** se la stringa fosse "amaca" la funzione dovrebbe restituire la stringa "ammaaaccccaaaaa".

- **B\_Ex2(m) (8 punti)** Completare la funzione Python **B\_Ex2(m)** che riceve in ingresso una matrice m e restituisce una nuova matrice che rappresenta la matrice triangolare superiore estratta da m. Se la matrice m non è quadrata (o vuota), la funzione dovrà restituire la matrice vuota.

**Esempio:** Se la matrice m è  $\begin{pmatrix} 2 & 2 \\ 3 & 3 \end{pmatrix}$  allora la matrice risultato sarà  $\begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix}$

Se invece la matrice m è  $\begin{pmatrix} 2 & 2 & 1 \\ 3 & 3 & 2 \end{pmatrix}$  allora la matrice risultato sarà quella vuota ([]) in Python).

- **B\_Ex3(file) (8 punti)** Implementare la funzione **B\_Ex3(file)** che prende in ingresso il nome di un file, contenente testo privo di punteggiatura. La funzione deve restituire un *dizionario* avente come chiavi le parole (*in lettere minuscole*) che compaiono in **file** e come valori i corrispondenti numeri di occorrenze. Eventuali maiuscole devono essere ignorate, per cui ad esempio “Casa” e “casa” vanno considerate la stessa parola.

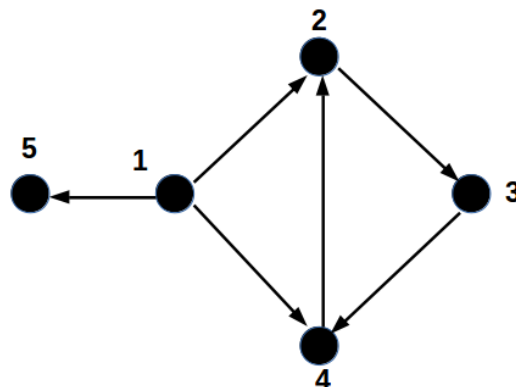
**Esempio:** si supponga che **file** contenga il testo:

“Sopra la panca la capra campà sotto la panca la capra muore”

La funzione deve restituire il dizionario {"sopra": 1, "la": 4, "panca": 2, "capra": 2, "campà": 1, "sotto": 1, "muore": 1}.

**Nota:** si ricordi che, data una stringa, il metodo lower restituisce una copia della stringa con caratteri minuscoli.

- **B\_Ex4(g, u, v) (8 punti)** Implementare la funzione **B\_Ex4(g, u, v)** che (tra gli altri parametri) riceve in ingresso un dizionario **g** che rappresenta un grafo *diretto*, i cui nodi hanno etichette *interi*. I parametri **u** e **v** sono nodi del grafo. La funzione deve restituire la somma delle etichette dei nodi che sono raggiungibili sia da **u** che da **v**. Si assuma che un nodo sia sempre raggiungibile a partire da se stesso.



Ad esempio, se il grafo fosse quello in figura, avremmo  $g = \{1: [4, 2, 5], 2: [3], 3: [4], 4: [2], 5: []\}$ . Se **u** fosse il nodo 1 e **v** il nodo 3 la funzione dovrebbe restituire 9 (i nodi 3, 4 e 2 sono raggiungibili da entrambi), mentre se **u** fosse il nodo 1 e **v** fosse il nodo 5, la funzione dovrebbe restituire il valore 5 (soltanto 5 è raggiungibile a partire da entrambi).