

10.07.2017

# 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 **B\_Ex1(s)** che prende come input una stringa s e restituisce la posizione del minimo carattere all'interno della stringa *secondo l'ordinamento definito da Python*. Se s contiene più volte il carattere massimo allora la funzione restituisce la posizione maggiore. Se s è la stringa vuota allora la funzione deve restituire -1.

**Esempio:** Se s='ABCabc' allora la funzione deve restituire 0, poiché il carattere minimo in questo caso è 'A' e compare una sola volta, in posizione 0.

- **B\_Ex2(s)** Scrivere una funzione che riceve in ingresso una stringa s e calcoli la lunghezza della più lunga sequenza che *a partire dall'inizio* (posizione 0) si ripete almeno 2 volte. Ad esempio, se la stringa s = "aafbcaadb" allora la funzione deve restituire 2 perchè la sequenza "aa" che parte dalla posizione 0 si ripete 2 volte e nessuna sequenza più lunga si ripete almeno 2 volte.
- **B\_Ex3(file) (8 punti)** Implementare la funzione Python **B\_Ex3(file)**, che riceve in ingresso il nome di un file contenente testo privo di punteggiatura. La funzione deve restituire la parola la cui frequenza nel testo è massima. Eventuali maiuscole devono essere ignorate, per cui ad esempio "Casa" e "casa" vanno considerate la stessa parola.

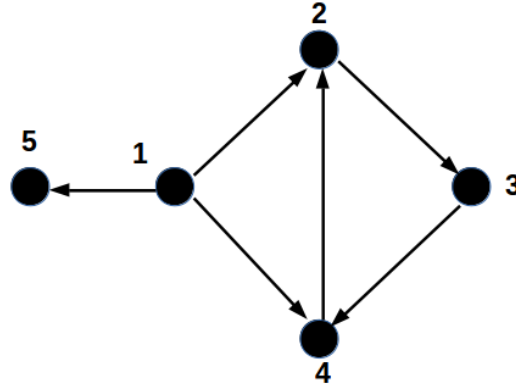
**Esempio:** se il testo fosse il seguente:

"Campa la capra sopra la panca camp"

la funzione dovrebbe restituire la "campa".

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

- **B\_Ex4(g, u) (8 punti)** Implementare la funzione **B\_Ex4(g, u)** che (tra gli altri parametri) riceve in ingresso un dizionario **g** che rappresenta un grafo *diretto*. Il parametro **u** è un nodo del grafo. La funzione deve restituire la lista *ordinata* dei nodi del grafo che *non hanno u come vicino*.



Ad esempio, se il grafo fosse quello in figura, avremmo  $\mathbf{g} = \{1: [4, 2, 5], 2: [3], 3: [4], 4: [2], 5: []\}$ . Se **u** fosse il nodo 2, la funzione dovrebbe restituire la lista  $[2, 3, 5]$ . Si noti che in questo caso 2 non ha se stesso come vicino, ma in generale ciò potrebbe accadere.