

05.05.2017

# 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.

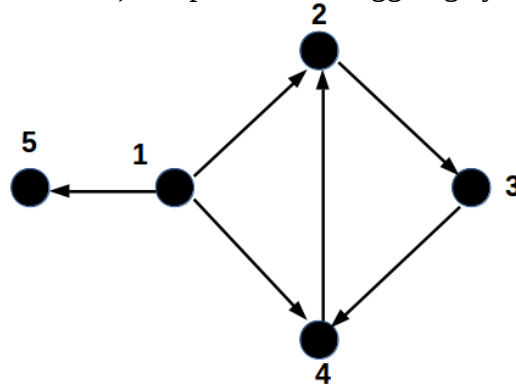
#### Esercizi

- **B\_Ex1(l,n) (8 punti)** Scrivere una funzione che riceve in ingresso una lista di interi **l** e un intero **n** e restituisce la lunghezza della sequenza più lunga di interi consecutivi di **l** che sono maggiori od uguali ad **n**. Ad esempio, se **l** vale [15, 15, 80, 21, 6, 5, 6, 21] e **n** vale 15 la funzione deve restituire 4.
- **B\_Ex2(file\_in)** Scrivere una funzione che prende in ingresso un **file\_in** che contiene **n** interi per ogni riga separati da ';' e restituisce una lista i cui elementi corrispondono alla somma dei numeri presenti su ogni riga del file. Ad esempio, se le righe del file sono "3;4;5" e "9;2;6", la funzione deve restituire la lista contenente 12 e 17.
- **B\_Ex3(m, c) (8 punti)** Completare la funzione Python che riceve in ingresso una matrice di stringhe **m** e un carattere **c** e restituisce l'indice della colonna **j** in cui il carattere **c** occorre con la frequenza minima (maggiore di 0).  
Se il carattere **c** non presente in alcuna colonna di **m**, allora la funzione deve restituire il valore -1. Se il carattere **c** appare su più colonne con la stessa frequenza minima, allora la funzione deve restituire l'indice più grande tra questi.

**Esempio:** Se la matrice **m** è  $\begin{pmatrix} \text{Ciao} & \text{Bye} \\ \text{Casa} & \text{Canale} \end{pmatrix}$  e il carattere **c** è "b" allora la funzione deve

restituire 1 mentre se il carattere **c** è "z" allora la funzione deve restituire -1.

- **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*, i cui nodi hanno etichette *intere*. Il parametro **u** è un nodo del grafo. La funzione deve restituire la lista *ordinata* dei nodi del grafo che *non sono raggiungibili da u*. Si ricordi che un nodo *y* è raggiungibile a partire da un nodo *x* se esiste un cammino (diretto nel nostro caso) che parte da *x* e raggiunge *y*.



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, la funzione dovrebbe restituire la lista [], perché tutti i nodi sono raggiungibili dal nodo 1. Al contrario, se **u** fosse il nodo 3, la funzione dovrebbe restituire la lista [1, 5].

*Si consideri che il nodo di partenza è raggiungibile per definizione.*