

16.02.2015

Compito d'esame di Fondamenti Informatica I (modulo Python) Compito C

Istruzioni (leggere attentamente)

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

Registrazione dei dati dello studente

Prima di iniziare il compito, cliccare su `Risorse` del `Computer`, aprire prima la cartella (di rete) che trovate nella finestra e poi la cartella `Esame`. 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 `A_ExN.py` (dove `N` è il numero dell'esercizio) con lo scheletro della soluzione.

Per testare un esercizio **DOVETE** usare il programma `TestEx.pyc`

Si consiglia, in ogni caso, prima di usare il programma `TestEx.pyc` di verificare la correttezza sintattica della vostra soluzione premendo `F5`.

E' possibile consultare la documentazione ufficiale del linguaggio Python (disponibile sul desktop cliccando su `Doc_Python3`), ma **non è possibile usare libri o appunti**.

In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

Esercizi

- **C_Ex1(s) (6 punti)** scrivere una funzione che riceve in ingresso una stringa `s` e calcoli la lunghezza della più lunga sequenza di caratteri diversi (ovvero in cui ciascun carattere compare solo una volta) **a partire dal carattere di indice 0**. Ad esempio se la stringa `s` vale `'acdfeca'` allora la funzione deve restituire 5, poiché la sequenza `'acdfc'` è lunga 5 e non contiene nessun carattere ripetuto.
- **C_Ex2(l1, l2) (6 punti)** scrivere una funzione che riceve in ingresso due liste `l1` e `l2` di numeri interi e verifichi se gli elementi della lista `l1` siano sempre esattamente maggiori degli elementi corrispondenti della lista `l2` (ovvero nelle stesse posizioni). Ad esempio, se `l1` vale `[5, 7, 9, 3, 11]` e `l2` vale `[3, 6]` la funzione deve restituire `True` in quanto sia nella posizione 0 (5 e 3) che nella posizione 1 (7 e 6) l'elemento di `l1` è maggiore del corrispondente di `l2`. Si faccia attenzione che le 2 liste possono essere di lunghezza diversa e che non si possono confrontare elementi che siano fuori dal range di una delle liste.
- **C_Ex3(file) (9 punti)** scrivere una funzione che prende in ingresso un file contenente il resoconto di compravendite in borsa. La prima riga del file ha il formato:

```
importoIniziale1;nome1;importoIniziale2;nome2;.....importoInizialen;nomen
```

in cui si specifica il credito disponibile per ogni investitore. Le righe successive contengono il resoconto delle varie compravendite, specificando in ognuna quanto ogni investitore ha guadagnato e perso nel seguente formato:

```
variazione1;variazione2;.....;variazionen
```

dove la variazione può essere positiva o negativa. Quando un investitore dopo una sessione di compravendita finisce il suo credito (bilancio ≤ 0) non può più effettuare compravendite e deve essere messo nella lista degli “investitori a secco” e in tutte le compravendite successive il suo guadagno sarà sempre 0. Leggere il file e costruire la lista ordinata (in ordine di esaurimento del credito, assumendo per semplicità che in ogni sessione venga eliminato al massimo un investitore) degli investitori a secco. Ad esempio se il **file** contiene

1000;Paolo;500;Marco;1000;Anna;300;Giorgio

-100;+100;-1500;+1000

-500;+700;0;-200

-400;+1000;0;-600

0;+1200;0;-500

la funzione deve restituire: ['Anna', 'Paolo', 'Giorgio'] poiché Anna finisce il credito alla prima compravendita, Paolo alla terza e Giorgio alla quarta.

- **C_Ex4(file1,file2) (9 punti)** scrivere una funzione che prende in ingresso due file **file1** e **file2**, dove **file1** contiene un insieme di offerte di viaggio per pool car (più persone si mettono d'accordo di viaggiare sulla stessa macchina per dividere i soldi della benzina):

NomeAutista;CittàPartenza;CittàArrivo;PostiLiberi

Il file **file2** contiene un insieme di ricerche di passaggi nel seguente formato:

NomeViaggiatore;CittàPartenza;CittàArrivo

Assumendo che per ciascun percorso esista una sola offerta, il programma deve restituire una lista, **ordinata alfabeticamente**, con i nomi degli autisti che, potenzialmente, possono riuscire a riempire la macchina. Ad esempio, se file1 è il seguente:

Paolo;Roma;Milano;3

Giovanni;Roma;Napoli;2

Anna;Firenze;Sondrio;3

E **file2** contiene:

Luca;Roma;Milano

Mario;Roma;Milano

Sandro;Roma;Milano

Sergio;Roma;Napoli

la funzione deve restituire:

['Paolo']