

Compito d'esame -- 20 gennaio 2020 -- Compito C

Istruzioni (leggere attentamente)

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

Registrazione dei dati dello studente: PRIMA DI INIZIARE, eseguite (con un doppio click sull'icona) il programma `RegistraStudente` che si trova nella cartella `Esame`. Inserite (separatamente) *Numero di Matricola*, *Cognome* e *Nome* seguendo le istruzioni che compaiono sul terminale, e confermate i dati che avete inserito. Il programma genera il file `studente.txt` che contiene Matricola, Cognome e Nome su tre righe separate (nell'ordine indicato). Il file `studente.txt` non deve essere modificato manualmente. Verificate che i dati nel file `studente.txt` siano corretti. In caso di errore potete rieseguire il programma `RegistraStudente`.

Svolgimento degli esercizi: Per ogni esercizio avete una cartella `EsercN` che contiene un file dal nome `C_ExN.py` (dove **N** è il numero dell'esercizio) con lo scheletro della soluzione. Questo file incorpora un codice di test che proverà la vostra soluzione per un certo numero di possibili dati in input. Aprite il file con IDLE e modificate SOLO il contenuto della funzione. Eseguendo il file `.py` si otterrà il responso dei test sulla console. Non spostate i file dalla loro posizione e non create nuovi file. **Si noti che per la correzione verranno usati insieme i 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

- **C_Ex1(s1,s2)** Scrivere una funzione che riceve in ingresso due stringhe **s1** e **s2** (dove **s2** è una stringa tale che ogni carattere compare al massimo una volta), e restituisce la stringa ottenuta da **s1** spostando tutti i caratteri che compaiono anche in **s2** all'inizio della stringa, ma rispettando l'ordine di occorrenza in **s2**. Ad esempio, se **s1** = "allegria" e **s2** = "jeah", la funzione deve restituire "eaallgri". Se invece **s1** = "allegria" e **s2** = "" la funzione deve restituire "allegria".
- **C_Ex2(m):** Una matrice quadrata si dice essere un "quadrato magico" se la somma dei numeri presenti in ogni riga, in ogni colonna, e in entrambe le diagonali, restituisce sempre lo stesso risultato; tale intero viene denominato "costante magica". Scrivere una funzione che prende in ingresso una matrice QUADRATA **m** di numeri interi positivi, rappresentata come lista di liste, e, se **m** è un quadrato magico, allora restituisce la tupla (`True`, **X**), dove **X** è la costante magica, altrimenti restituisce la tupla (`False`, **Y**), dove **Y** è la somma della diagonale principale.

Ad esempio, data la matrice

2	7	6
9	5	1
4	3	8

la funzione deve restituire la tupla (`True`, 15).

Invece, data la matrice

1	0	0
1	5	0
0	0	-1

la funzione deve restituire la tupla (`False`, 5).

- **C_Ex3(file)** In un file di testo sono presenti all'interno parole (stringhe alfanumeriche) e indirizzi email. Nel file non ci sono simboli di punteggiatura e parole ed indirizzi email sono separate da spazi o andate a capo. Assumete che gli indirizzi email contenuti nel testo siano SEMPRE sintatticamente corretti (semplificando, questo vuol dire che contengono esattamente una occorrenza del simbolo @, preceduta e seguita da caratteri alfanumerici e punti; c'è sempre almeno un carattere alfanumerico sia prima che dopo la @, ed almeno un punto dopo la @; inoltre, due punti non sono mai consecutivi, la @ ed un punto non sono mai consecutivi, l'email non termina con un punto). Esempi di email che NON possono comparire nel testo sono @gmail.com, mario.@gmail.com, alberto@, anna..rossi@yahoo.it. Potete anche assumere che gli indirizzi email NON sono mai spezzati su più righe. L'informazione fondamentale che dobbiamo estrarre è il **dominio**, che è sempre la sottostringa alfanumerica dell'email che precede l'ultimo punto.

Dovete scrivere una funzione che prende in input il nome del file e restituisce un dizionario che abbia come chiavi i domini estratti dalle email e come valore per ogni chiave l'insieme dei *prefissi* delle email, cioè la sottostringa dell'email che precede la @. Per risolvere questo problema potete usare le espressioni regolari, ma il loro uso non è indispensabile, scegliete voi liberamente se usarle o meno. Ad esempio, se il file è:

```
marco.marco@gmail.com indirizzo email assegnato a marco mentre anna si raggiunge con
anna@diag.uniroma1.it
```

Allora il risultato sarà {'gmail':{'marco.marco'}, 'uniroma1':{'anna'}}

- **C_Ex4(file):** Si consideri un file csv che contiene i risultati delle estrazioni del lotto su *una* determinata ruota relativi ad alcune. Si ricorda che per ogni ruota del lotto c'è una sola estrazione a settimana e che ad ogni estrazione vengono estratti esattamente 5 numeri *distinti* compresi fra 1 e 90. Scrivere la funzione che prende in input il nome del file e restituisce la lista ORDINATA IN MODO CRESCENTE E SENZA RIPETIZIONI dei **numeri interi** che sono stati estratti più volte nelle estrazioni contenute nel file. Se il file in input è *vuoto* deve restituire la *lista vuota*. Ad esempio, se il file in input contiene i seguenti dati

```
14, 50, 25, 47, 43
26, 82, 69, 75, 66
73, 50, 81, 45, 52
12, 14, 80, 90, 3
```

La funzione deve restituire [14, 50], dato che entrambi sono stati estratti due volte, e due è il massimo numero di volte che un numero è stato estratto.