

Autovalutazione – Python – 2 Funzioni e Strutture Dati Complesse

Introduzione

Gli esercizi proposti in questo foglio costituiscono uno strumento per valutare la propria preparazione. La soluzione dell'esercizio sarà discussa assieme ai docenti.

Indicazioni importanti

- La descrizione dei singoli esercizi è preceduta da una parte comune, leggetela attentamente
- Gli esercizi vi richiederanno di implementare uno script python

Esercizi: parte comune

Aperte il file *formula1.py* (che potete scaricare dalla piattaforma di e-learning del corso), troverete due strutture dati precaricate, con tutti i risultati del mondiale piloti di Formula 1 della stagione 2008, di cui qua sotto riportiamo uno stralcio. Una struttura dati precaricata è una struttura dati a vostra disposizione dentro lo script, senza che dobbiate preoccuparvi di eseguire l'input dei dati.

```
piloti=("Hamilton","Massa","Raikkonen","Kubica","Alonso","Heidfeld","Kovalainen",
       "Vettel","Trulli","Glock")

ord_arrivo=[
    ("Melbourne","Australia","1","0","8","0","4","2","5","0","0","0"),
    ("Sepang","Malesia","5","0","1","2","8","6","3","0","4","0"),
    ...
]
```

La tupla *piloti* è l'elenco di 10 piloti in gara, la lista *ord_arrivo* è formata da tuple composte da 12 elementi. I primi 2 elementi rappresentano la città e lo stato dove si è svolto il gran premio, i 10 interi successivi rappresentano l'*ordine di arrivo* di ciascun pilota nella singola gara (0 significa che il pilota non è arrivato tra i primi 8). Attenzione, i 10 interi con le informazioni sull'ordine di arrivo dei piloti, sono memorizzati con lo stesso ordine con cui sono memorizzati i nomi dei piloti nella tupla *piloti*. Per es., nel gran premio di Melbourne, Hamilton è arrivato primo mentre Raikkonen è arrivato ottavo.

Al termine di un gran premio, vengono assegnati i punti ai piloti in base all'ordine di arrivo: 10 al vincitore, 8 al secondo, 6 al terzo, e poi a scalare di 1 punto fino all'ottavo.

B1. Formula1. Esercizio puntipilota.

Implementate una funzione in linguaggio python con la struttura indicata qua sotto:

```
def puntiPilota(nome):
```

La funzione riceve in ingresso il nome di un pilota (l'utente che richiamerà la funzione dovrà passare come nome del pilota un nome identico ad uno dei nomi presenti nella lista precaricata) e deve restituire una lista (formata da numeri interi) contenente: il totale dei punti, il numero di vittorie (cioè quante volta il pilota è arrivato primo) e il numero dei podi del pilota (un pilota sale sul podio quando arriva al 1°, al 2°, oppure al 3° posto). La lista restituita dovrà essere composta solo da numeri interi e i numeri dovranno rispettare l'ordine seguente: [punti, n_vittorie, n_podi]. Per esempio `puntipilota("Hamilton")` dovrà restituire la seguente lista: [98, 5, 10].

Potete controllare i risultati prodotti dal vostro algoritmo con i risultati riportati in questo sito web: http://it.wikipedia.org/wiki/Formula_1_stagione_2008

Potete scrivere il vostro codice nello script *formula1.py* che potete scaricare dalla piattaforma di e-learning.

[Continua nel foglio successivo]

B2. Formula1. Esercizio classificagp.

Implementate una funzione in linguaggio python con la struttura indicata qua sotto:

```
def classificaPiloti():
```

La funzione non riceve alcun parametro in ingresso e deve restituire un dizionario formato da coppie chiave valore, dove la chiave è una stringa che contiene il nome del pilota, mentre il valore associato alla chiave è un intero che rappresenta il totale dei punti che il pilota ha conseguito alla fine del campionato mondiale. Come chiavi dovranno essere usate esattamente le stesse stringhe che appaiono nella variabile piloti (se nella variabile piloti appare “Alonso”, usate quella stringa come chiave, non usate altre stringhe del tipo “Fernando Alonso” oppure “F. Alonso” ...). Potete controllare i risultati prodotti dal vostro algoritmo con i risultati riportati in questo sito web: http://it.wikipedia.org/wiki/Formula_1_stagione_2008
Potete scrivere il vostro codice nello script formula1.py che potete scaricare dalla piattaforma di e-learning.

Nel file è presente un'ulteriore struttura dati, il dizionario pil2cost, che contiene un insieme di coppie chiave valore, dove la chiave è il nome del pilota e il valore è il nome del costruttore per il quale il pilota corre.

B3. Formula1. Esercizio classificacostruttori.

Implementate una funzione in linguaggio python con la struttura indicata qua sotto:

```
def classificaCostruttori():
```

La funzione non riceve alcun parametro in ingresso e deve restituire un dizionario formato da coppie chiave valore, dove la chiave è una stringa che contiene il nome del costruttore, mentre il valore associato alla chiave è un intero che rappresenta il totale dei punti che il costruttore ha conseguito alla fine del campionato mondiale. I punti conseguiti da un costruttore sono la somma dei punti che i piloti che corrono per il costruttore hanno conseguito durante l'anno. Fate attenzione, nella struttura dati precaricata non sono presenti i dati di tutti i piloti che hanno partecipato alle corse della stagione 2008, per questo motivo, i risultati ottenuti potrebbero differire rispetto a quanto riportato in: http://it.wikipedia.org/wiki/Formula_1_stagione_2008. Potete scrivere il vostro codice nello script formula1.py che potete scaricare dalla piattaforma di e-learning. Nota bene: nelle strutture dati precaricati mancano i dati di alcuni dei piloti delle squadre Renault e Toro Rosso, per questo motivo i dati calcolati dalla funzione per queste due squadre non corrispondono con i dati reali.

Fase di test.

Negli esercizi proposti, avete avuto a che fare con liste o dizionari contenenti un numero elevato di dati. Se necessario, per semplificare la fase di test del vostro codice, all'inizio non lavorate direttamente su strutture dati di grosse dimensioni, ma utilizzando i commenti su più righe (cioè i tripli apici o le triple virgolette) escludete porzioni di dati in modo da lavorare su una struttura dati di dimensioni ridotte (per esempio con solo 4 o 5 elementi). Una volta che il vostro programma inizierà a funzionare correttamente con le strutture dati “ridotte”, levate i commenti e lavorate sui dati originali.

Prestate attenzione perché i valori contenuti nelle liste e nei dizionari potranno essere a loro volta elementi complessi come per esempio delle tuple.

Buon lavoro

I docenti del corso