

Prima parte

Sviluppare un programma Python per analizzare i risultati dello scorso campionato di calcio in alcune nazioni europee. Il file Excel **all-euro-data-2016-2017.xls** contiene i risultati dei campionati di calcio europei della stagione 2016/2017. Dei ventidue campionati considerare ai fini del progetto: E0, SC0, D1, SP1, I1, F1, N1, B1, P1, T1, G1.

Di tutti i campi presenti in ogni foglio Excel, quelli da considerare per il progetto sono:

- Div = Campionato (è una sigla, associare un nome)
- Date = Giorno incontro (gg/mm/aa)
- HomeTeam = Nome squadra locale
- AwayTeam = Nome squadra ospite
- FTHG = Goal squadra locale alla fine della partita
- FTAG = Goal squadra ospite alla fine della partita
- FTR = Risultato finale (H=Vittoria, D=Pareggio, A=Sconfitta)
- HTHG = Goal squadra locale primo tempo
- HTAG = Goal squadra ospite primo tempo
- HTR = Risultato primo tempo (H=Vittoria, D=Pareggio, A=Sconfitta)

I dati nei fogli associati ai campionati indicati in precedenza devono essere organizzati in opportune strutture dati per rispondere alle seguenti richieste:

1. Dato un campionato, stampare l'elenco delle squadre del campionato.
2. Dati una giornata e un campionato, stampare la classifica per la giornata indicata. Per ogni squadra indicare anche le partite giocate (una partita può essere stata rinviata).
3. Dati una giornata e un campionato, stampare la classifica per la giornata indicata considerando i risultati che si riferiscono al primo tempo. Per ogni squadra indicare anche le partite giocate.
4. Date una giornata e una squadra, stampare gli ultimi cinque risultati per la squadra indicata (è sufficiente stampare una sequenza di W/D/A).
5. Dato un giorno, stampare i risultati di tutte le eventuali partite giocate il giorno indicato.
6. Dati una giornata e un intero k, stampare le k squadre che hanno segnato più goal.
7. Dati una giornata e un intero k, stampare le k squadre che hanno subito meno goal.
8. Dati una giornata e un intero k, stampare le k squadre con la migliore differenza reti.
9. Dati una giornata e un campionato, stampare la squadra con il maggior numero di vittorie, la squadra con il maggior numero di vittorie in casa, la squadra con il maggior numero di vittorie in trasferta.

Per i punti 6, 7, e 8 le k squadre devono essere considerate tra tutti i campionati.

Per ognuna delle richieste precedenti, valutare la complessità di tempo e di spazio della soluzione proposta. Le strutture dati utilizzate possono essere utilizzare per rispondere anche a richieste non previste dal progetto?

Per determinare la giornata del campionato potrebbe essere utilizzata la seguente tecnica: dato il numero N di squadre nel campionato, una giornata è determinata da $N/2$ righe consecutive nel foglio associato al campionato. Se una partita è stata annullata e poi è recuperata in seguito, l'approccio precedente non funziona (ad esempio, alla terza giornata dello scorso campionato di Serie A, Genoa-Fiorentina è stata rinviata per pioggia). Come si può risolvere questo problema? L'applicazione sviluppata deve tener conto di questa eventualità, ad esempio, la classifica alla terza giornata di Serie A indicherebbe Genoa a 6 punti e Fiorentina a 3 punti, con due partite giocate da entrambe le squadre.

Allegare al progetto una relazione che illustri le strutture dati utilizzate e la loro funzione.

Seconda parte

Il metodo **count** della classe Python **str** restituisce il numero di occorrenze non sovrapposte di un pattern in una stringa. Ad esempio, `'ababa'.count('aba')` restituisce 1, mentre `'abababa'.count('aba')` restituisce 2.

Adattare l'algoritmo di pattern-matching di Knuth-Morris-Pratt per implementare la funzione **count_kmp**(T,P) che, come **count**, restituisce il numero di occorrenze non sovrapposte del pattern P nella stringa T .

Progettare un insieme di test per valutare l'efficienza della soluzione proposta.

Chi vuole sviluppare un'interfaccia grafica può far riferimento a **Tk**, <http://www.tkdocks.com/>