

Programmazione I

Prova di Programmazione – 26 febbraio 2015 – 2 ore 15 min

Partendo dal frammento di codice fornito, realizzare un programma di gestione turni di lavoro, rappresentati mediante una sequenza di elementi, la *lista dei turni*, in cui ciascun elemento è costituito da una data e dalla lista dei nomi di coloro che sono di turno in quella data. Ciascun nome è semplicemente una parola. Le *date valide* sono nel formato *gg/mm*, assumendo implicitamente che l'anno sia il 2015, e con sia *gg* che *mm* rappresentati su due cifre (ad esempio 02/03 e non 2/3). All'avvio del programma, la lista dei turni è vuota. La lunghezza della lista dei turni non è nota a tempo di scrittura del programma, mentre la lunghezza massima dei nomi ed il numero massimo di turni per ogni data sono noti. Il programma deve fornire le seguenti funzionalità.

1. **carica_turni** Carica la lista dei turni da un file di testo dal nome predefinito, e dal formato che si ritiene più opportuno. Un esempio di contenuto di file potrebbe essere:
3
23/07 2 Valente Andreolini
28/07 1 Andreolini
05/09 3 Leoncini Colajanni Zanni
Assumere che il file non contenga mai due volte lo stesso nome per una stessa data e che le date siano tutte valide ed in ordine crescente. L'eventuale precedente lista dei turni è persa.
2. **stampa_turni** Stampa la lista dei turni su *stdout*. con il seguente formato, esemplificato nel caso della lista di cui all'esempio al punto 1:
23/07 Valente, Andreolini
28/07 Andreolini
05/09 Leoncini, Colajanni, Zanni
3. **converti_data(stringa)** Possibile funzione di supporto per il punto seguente. Se e solo se la stringa **stringa** contiene esattamente e solamente una data valida, ritorna il numero del giorno corrispondente all'interno dell'anno, ossia un numero compreso tra 1 e 365. Altrimenti ritorna 0. In questa funzionalità la stringa non si legge da *stdin*. Ad esempio, la funzione ritorna 102 per la stringa 12/04 e 0 per le stringhe 32/03, 22 oppure a/7. Si ottiene il punteggio massimo se il calcolo è effettuato a costo costante rispetto al valore del mese (eventualmente grazie all'esecuzione di qualche altra funzione all'avvio del programma). Potrebbe essere utile considerare l'esistenza della funzione *isdigit(c)* che ritorna vero se il valore *c* è il codice di una cifra compresa tra '0' a '9', ed il fatto che può essere necessario includere *ctype.h* per utilizzare tale funzione.
4. **cambia_data(idx)** Legge una riga intera da *stdin* e la memorizza in una stringa, dopodiché, solo se tale riga contiene esattamente e solamente una data valida, sostituisce la data per il turno di indice **idx** con la nuova data letta da *stdin*. Ad esempio, se **idx** è uguale a 1, e legge la riga 05/08 da *stdin*, modifica la data nella seconda riga dell'esempio del punto 2 da 28/07 a 05/08. Si ottiene il punteggio massimo se la funzionalità riordina la lista dei turni se necessario a mantenerla in ordine, assumendo che la lista sia ordinata prima dell'invocazione della funzionalità stessa.

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'inserimento di dati in formato errato e di messaggi troppo lunghi da *stdin*.

REGOLE

- Si può utilizzare ogni genere di manuale o di materiale didattico di altra natura
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1 e 2. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
 - a) il programma è perfettamente funzionante in ogni sua parte
 - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati