

**Programmazione I**  
**Prova di programmazione – 29 Gennaio 2020 – 2 ore**

Partendo dal frammento di codice fornito, realizzare un programma di gestione di un sondaggio. Il sondaggio serve a stabilire il giorno in cui effettuare un incontro, all'interno di una settimana lavorativa. Tale settimana è di soli cinque giorni, dal lunedì al venerdì. Il programma non conosce, e non gestisce, le date in cui cadono i giorni di tale settimana. Gli interessati all'incontro si iscrivono al sondaggio, comunicando il proprio nome (una sola parola, di lunghezza massima definita a tempo di scrittura del programma) ed i giorni in cui sono disponibili (eventualmente nessun giorno). Il numero massimo di partecipanti al sondaggio non è noto a tempo di scrittura del programma. All'avvio del programma il sondaggio è vuoto. Il programma fornisce le seguenti funzionalità.

1. **aggiungi\_partecipante(nome, giorni)** Iscrive il partecipante di nome **nome** al sondaggio, con le disponibilità dichiarate nella stringa **giorni**. La stringa **giorni** è una sequenza di cinque caratteri a scelta tra asterischi e trattini, senza spazi nel mezzo. Un asterisco o trattino in posizione *i*-esima significa che per il giorno *i*-esimo il partecipante è, rispettivamente, disponibile o non disponibile. Ad esempio, la stringa **--\*-\*** comunica che il partecipante è disponibile solo al mercoledì ed al venerdì. Si ottiene il punteggio massimo se si realizza questa funzionalità con costo ammortizzato inferiore a  $O(N)$ , senza preoccuparsi di ottimizzare anche la quantità di memoria occupata. Per costo ammortizzato intendiamo il costo di  $N$  inserimenti, diviso  $N$ .
2. **stampa\_sondaggio** Stampa l'attuale contenuto del sondaggio, nel formato mostrato attraverso l'esempio seguente  
**L M M G V Nome**  
**\* - \* \* - Paolo**  
**- - \* - \* Roberto**  
**\* - \* \* \* Michele**  
Se il sondaggio è vuoto, allora stampa solo la riga di intestazione.
3. **salva\_sondaggio** Salva il sondaggio in un file **binario** dal nome predefinito. Ricordarsi che per salvare, attraverso una *write()*, il contenuto di un array dinamico memorizzato in un oggetto di tipo struttura, non è corretto passare l'indirizzo dell'oggetto di tipo struttura.
4. **carica\_sondaggio** Carica il sondaggio dal file. L'eventuale precedente sondaggio è perso.
5. **trova\_giorno\_migliore** Trova il **primo** giorno in cui il massimo numero di partecipanti è disponibile. Nell'esempio di cui al punto 2, tale giorno sarebbe il mercoledì. Per questa funzionalità, il programma dovrà stampare il nome del giorno, con la lettera finale non accentata, su una riga separata. Quindi, per il solito esempio, il programma dovrebbe stampare **mercoledì**. Se nessuno dei partecipanti è mai disponibile, allora il programma deve stampare **nessuno**.

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

Per compatibilità con il tester, configurare una lunghezza massima dei nomi pari almeno a 20.

---

#### REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1 e 2. Se si svolgono solo tali punti, il programma deve essere perfettamente funzionante. 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
  - c) sono state seguite eventuali altre indicazioni presenti nella traccia in merito al voto finale