

Programmazione I
Prova di programmazione – 23 Febbraio 2022 – 2 ore

Partendo dal frammento di codice fornito, realizzare un gestore di turni di lavoro. Il programma gestisce un elenco di lavoratori, in cui ciascun lavoratore è identificato da un nome (una sola parola, di al massimo 10 caratteri). I turni riguardano una sola settimana di lavoro, dal lunedì alla domenica. Per ciascun lavoratore e giorno della settimana, il programma memorizza se tale lavoratore è di turno o di riposo, ossia se deve lavorare o meno, in tale giorno. All'avvio del programma, l'elenco di lavoratori è vuoto. Il programma fornisce le seguenti funzionalità.

1. **[+1] inizializza_elenco(N)** Inizializza l'elenco a contenere N lavoratori e legge da *stdin* il nome di ciascuno di tali lavoratori. Tutti i lavoratori sono di riposo in tutti i giorni. Assumere che l'utente non inserisca mai due lavoratori con lo stesso nome. L'eventuale precedente elenco è perso, assieme ai turni dei lavoratori appartenenti a tale elenco. Si ottiene il punteggio aggiuntivo se si evitano deallocazioni e riallocazioni di memoria non necessarie.
2. **stampa_turni** Stampa le iniziali dei giorni della settimana, seguite dai turni dei lavoratori, con il formato mostrato nel seguente esempio (asterisco vuol dire di turno, trattino vuol dire non di turno)

```
LMMGVSD
**-----* Paolo
--*---** Sonia
-*---** Rita
```

3. **[2] trova_indice(n)** Se il lavoratore di nome n è presente, ritorna la sua posizione nell'elenco. Le posizioni partono da 0. Se il lavoratore non è presente, ritorna -1.
4. **[3] aggiorna_turni(n, t)** Aggiorna i turni del lavoratore di nome n . Il parametro t contiene i nuovi turni, sotto forma di stringa di sette caratteri. Un carattere '*' oppure '-' nella posizione i -esima indica che, nel giorno i -esimo della settimana, il lavoratore n deve essere, rispettivamente, di turno oppure di riposo.
5. **[2] salva_turni** Salva i turni in un file di testo.
6. **[2, +1] carica_turni** Carica i turni dal file di testo. Il precedente contenuto dell'archivio è perso. Si ottiene il punteggio aggiuntivo se si evitano deallocazioni e riallocazioni di memoria non necessarie.
7. **[3, +2] minimizza_turni()** Elimina turni in maniera tale che ciascuno dei lavoratori svolga il numero minimo possibile di turni, col vincolo che rimanga presente un lavoratore nei giorni in cui era presente almeno un lavoratore in turno

```
LMMGVSD
*-----* Paolo
--*---*- Sonia
-*---*- Rita
```

Si ottiene il punteggio aggiuntivo se si realizza questa funzionalità a costo lineare rispetto al numero totale di possibili turni.

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*.

Fare stampare un accapo prima dell'indice al punto 3, per evitare che tale indice sia eventualmente stampato come parte di una riga preesistente, e quindi non riconosciuto dal tester.

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 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