

**Programmazione I**  
**Prova di programmazione – 18 Febbraio 2021 – 2 ore**

Partendo dal frammento di codice fornito, realizzare un programma per la gestione di una lotteria. Ogni biglietto viene creato mediante una operazione vendita, operazione in cui si fornisce il nome di un proprietario. Il biglietto è identificato da un numero intero strettamente positivo univoco, deciso all'atto della creazione. Il numero massimo di biglietti che possono essere venduti non è noto a tempo di scrittura del programma. Il nome di ciascun proprietario è una parola di lunghezza massima  $M$ , decisa a tempo di scrittura del programma. All'avvio del programma, non c'è ancora alcun biglietto venduto. Il programma fornisce le seguenti funzionalità.

1. **[+3] vendi\_biglietto(p, n)** Vende un nuovo biglietto per un proprietario di nome **p** (dove **p** è una parola non nulla), e con numero **n**. Il numero del biglietto deve essere univoco, mentre invece uno stesso proprietario può comprare più biglietti. Si ottiene il punteggio aggiuntivo se si realizza questa funzionalità a costo ammortizzato  $O(1)$  rispetto al numero di biglietti venduti.
2. **stampa\_biglietti** Stampa il nome del proprietario ed il numero del biglietto per ciascun biglietto venduto. Stampa i biglietti nell'ordine in cui sono stati venduti. Ad esempio:  
**Mario 231421**  
**Anna 98361**
3. **[3] salva\_biglietti** Salva l'insieme dei biglietti venduti in un file di testo.
4. **[3] carica\_biglietti** Carica l'insieme dei biglietti venduti da un file di testo. Il precedente insieme è perso.
5. **[2] estrai\_vincitore()** Estrae un biglietto a caso, e stampa il nome del proprietario ed il numero del biglietto. Oppure stampa **NESSUNO 0** nel caso in cui non vi sia alcun biglietto venduto. Ad esempio, dato lo stato mostrato nell'esempio al punto 2, questa funzionalità potrebbe stampare  
**Anna 98361**  
oppure  
**Mario 231421**
6. **[4] estrai\_primi\_n\_biglietti(n)** Estrae **n** biglietti a caso e, per ciascuno, stampa il nome del proprietario ed il numero del biglietto. Se vi sono meno di **n** biglietti venduti estrae e stampa un numero di biglietti pari al numero di biglietti venduti. Se non vi è alcun biglietto venduto, stampa **NESSUNO 0**. Si ottiene il punteggio massimo se il costo dell'estrazione di ciascun biglietto è  $O(N)$  rispetto al numero  $N$  di biglietti venduti.

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, il numero del biglietto ed il nome del proprietario vanno inseriti una sola volta per la funzionalità 1, anche in caso di fallimento della funzionalità. Inoltre, per la funzionalità 6 assicurarsi di stampare un *newline* prima della stampa dei biglietti, altrimenti eventuali caratteri precedenti, non seguiti da *newline*, andrebbero a fondersi con la riga contenente il primo biglietto.

---

#### 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