

**Programmazione I**  
**Prova di programmazione – 9 Febbraio 2022 – 2 ore**

Partendo dal frammento di codice fornito, realizzare un gestore di un archivio di prodotti. Ciascun prodotto è identificato da un codice, ossia un numero naturale da 1 a  $N$ , con  $N$  modificato a tempo di esecuzione del programma. Per ciascun prodotto si memorizza (solamente) quante unità sono presenti. All'avvio del programma, l'archivio è vuoto ed  $N=1$ . Il programma fornisce le seguenti funzionalità.

1. **[+1] inizializza\_archivio(N)** Inizializza l'archivio a contenere prodotti con codici compresi tra 1 ed  $N$ . L'eventuale precedente contenuto dell'archivio è perso. Si ottiene il punteggio aggiuntivo se si evitano deallocazioni e riallocazioni di memoria non necessarie.
2. **aggiungi\_prodotto(n)** Aggiunge all'archivio una unità per il prodotto di codice  $n$ . Ad esempio, se si invoca questa funzionalità con  $n = 3$ , su un archivio che non contiene alcun prodotto con codice 3, allora l'archivio conterrà poi un solo prodotto con codice 3. Se invece l'archivio conteneva già un prodotto con codice 3, allora ne andrà a contenere 2, e così via.
3. **[+2] stampa\_archivio** Stampa, in ordine di codice crescente, i soli prodotti presenti in archivio (ossia con un numero di unità maggiore di 0). Per ciascuno di tali prodotti, stampa il codice del prodotto ed il numero di unità presenti. Ad esempio:

```
3 5
16 3
20 37
```

Si ottengono i due punti aggiuntivi se si realizza questa funzionalità a costo lineare rispetto al solo numero di prodotti presenti, e non rispetto al numero  $N$  di codici possibili.

4. **[2] salva\_archivio** Salva l'archivio in un file di testo.
5. **[3, +1] carica\_archivio** Carica l'archivio 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.
6. **[2, +1 o +4] leggi\_elimina\_primo\_prodotto()** Ritorna il numero di unità del prodotto di codice più basso presente in archivio, ed elimina tale prodotto dall'archivio. Ritorna 0 sia se l'archivio non contiene alcun prodotto. Ad esempio, invocando questa funzionalità sull'archivio mostrato al punto 3, l'archivio diverrebbe

```
16 3
20 37
```

Si ottiene il punto aggiuntivo, oppure i quattro punti aggiuntivi, e si realizza questa funzionalità a costo lineare, oppure costante, rispetto al numero di prodotti presenti.

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*.

---

#### REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1, 2 e 3. 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