

# Programmazione I

## Prova di Programmazione – 5 febbraio 2016 – 2 ore 15 min

Partendo dal frammento di codice fornito, realizzare un programma per la visualizzazione di una successione  $S$  di valori interi. La lunghezza  $N$  della successione è stabilita a tempo di scrittura del programma. All'avvio del programma la successione contiene  $N$  zeri. Se utile, la sintassi per definire ed allocare una matrice dinamica, dal numero di colonne prefissato e pari ad  $N$ , è la seguente:

```
<tipo_elem> (*<identificatore>) [N] = new <tipo_elem>[<num_righe>] [N] ;
```

Inoltre, le seguenti due notazioni sono equivalenti nella definizione di parametri formali:

```
<tipo_elem> (*<identificatore>) [N]            <tipo_elem> <identificatore>[] [N]
```

Realizzare le seguenti funzionalità.

1. **inizializza\_successione()** Legge da *stdin* una successione di  $N$  valori interi (come scritto sopra,  $N$  è fissato a tempo di scrittura del programma), ed inizializza la successione  $S$  con la successione di valori letta da *stdin*. L'eventuale precedente contenuto della successione  $S$  è perso.
2. **stampa\_grafico1()** Stampa un grafico a barre dei valori della successione  $S$ , solo se i valori memorizzati nella successione sono tutti compresi tra 0 e 5 (estremi inclusi). Utilizza il carattere \* per disegnare le barre. Ad esempio, se  $N$  è uguale a 10, e la successione è [1, 2, 3, 3, 2, 1, 0, 0, 2, 1], allora stampa

```
**
 *  *  *
 *    *  *
      **
```

Non porsi alcun problema di occupazione della memoria.

3. **salva\_succ()** Salva la successione  $S$  su un file binario dal nome definito a tempo di scrittura del programma.
4. **carica\_succ()** Ricarica la successione  $S$  dal file. L'eventuale precedente contenuto è perso.
5. **stampa\_grafico2()** Identica alla funzionalità **stampa\_grafico1()**, ma senza vincoli sui valori memorizzati nella successione, e con l'aggiunta della stampa di una sequenza di trattini in corrispondenza del valore 0, solo nel caso in cui siano presenti anche valori negativi nella successione. Ad esempio, se la successione contiene i valori [1, 0, -1, -2, -2, -1, 0, 1, 0], stampa:

```
*          *
- * - - - - * - *
 *   **
      **
```

Porsi il problema di occupare la memoria minima possibile. Si ottiene il punteggio massimo se si riduce al minimo anche il numero di operazioni di gestione della memoria effettuate nel caso più fortunato.

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