

# Programmazione I

## Prova di Programmazione – 13 settembre 2019 – 2 ore 15 min

Partendo dal frammento di codice fornito, realizzare un programma per la visualizzazione di istogrammi rappresentanti una forma d'onda (*waveform*) audio. Una *waveform* è rappresentata come una sequenza temporale di un numero finito  $N$  di campioni, rappresentanti l'ampiezza della forma d'onda in ogni istante, e in ultima istanza memorizzata come una successione di  $N$  valori interi ( $N$  sia stabilito a tempo di scrittura del programma). All'avvio del programma la *waveform* è inizializzata con  $N$  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] ;
```

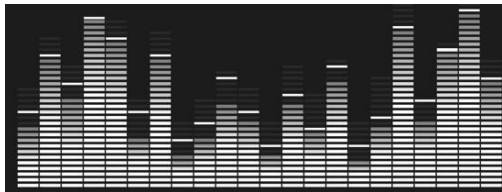
Inoltre, le seguenti due notazioni sono equivalenti nella definizione di parametri formali:

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

Realizzare le seguenti funzionalità:

1. **inizializza\_waveform ()** Legge da *stdin*  $N$  valori interi (come scritto sopra,  $N$  è fissato a tempo di scrittura del programma), utilizzati per inizializzare la successione di campioni rappresentante la *waveform*. L'eventuale precedente contenuto della *waveform* è perso.
2. **stampa\_istogramma ()** Stampa un istogramma a barre della *waveform*, solo se i valori dei suoi campioni sono tutti compresi tra 0 e 5 (estremi inclusi). Le barre sono piene (ovvero, non bisogna tracciare solo il profilo della *waveform*, ma tutta la sua area). Utilizza il carattere '=' per disegnare le barre. Es., se  $N=10$ , e la *waveform* è composta dai campioni [1, 2, 3, 3, 2, 1, 0, 0, 2, 1], stampa

```
3    ==
2    ===== =
1    ===== ==
0    =====
   0123456789
```



Non occorre stampare i valori sugli assi (qui mostrati a puro scopo illustrativo). Non porsi alcun problema di occupazione della memoria.

3. **salva\_waveform ()** Salva la successione dei campioni della *waveform* su un file di testo dal nome definito a tempo di scrittura del programma.
4. **carica\_waveform ()** Ricarica la *waveform* dal file. L'eventuale precedente contenuto è perso.

5. **stampa\_istogramma\_2 ()** Identica alla funzionalità **stampa\_istogramma ()**, ma senza vincoli sui valori assunti dai campioni, e con l'aggiunta della stampa di una sequenza di trattini in corrispondenza del valore 0, solo nel caso in cui siano presenti anche campioni di valore negativo. Ad esempio, se la *waveform* contiene i valori [1, 2, 3, 2, 1, 0, -2, -4, -2, 0], stampa:

```
3    =
2    ===
1    =====
0    -----
-1           ===
-2           ===
-3           =
-4           =
      0123456789
```

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.

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