

Programmazione I

Prova di programmazione – 14 Settembre 2018 – 2 ore

Partendo dal frammento di codice fornito, realizzate un programma per la stampa del grafico a barre di una sequenza di valori interi non negativi. All'avvio del programma, non esiste alcuna sequenza. Il programma deve fornire le seguenti funzionalità.

1. **reinizializza_sequenza(N)** Inizializza una sequenza di **N** valori, leggendo i valori da *stdin*. Il valore massimo di **N** non è noto a tempo di scrittura del programma.
2. **stampa_grafico_orizzontale(M)** Stampa il grafico a barre della sequenza, rappresentando ciascun valore con una barra orizzontale. Ogni batta è formata da una sequenza di caratteri **#**, e la lunghezza delle barre è scalata in maniera tale che la barra corrispondente al valore massimo sia lunga **M** caratteri. Ad esempio, se alla funzionalità è passato **M=5**, ed la sequenza di valori è (573, 857, 213, 25), stampa

#
3. **salva_sequenza** Salva la sequenza su di un file di testo dal nome predefinito.
4. **carica_sequenza** Carica la sequenza dal file. L'eventuale precedente sequenza è persa.
5. **stampa_barra_verticale(i, M)** Stampa il valore dell'*i*-esimo elemento della sequenza, tramite una barra verticale, ossia una sequenza verticale di caratteri **#**. La lunghezza della barra deve essere scalata dello stesso fattore per il quale deve essere scalata la barra relativa al valore massimo nella sequenza, affinché tale barra massima risulti lunga **M**. Ad esempio, dato la sequenza di esempio al punto 2, se *i=0* ed **M=5**, allora stampa:

```
#  
#  
#
```

6. **stampa_grafico_verticale(i1, i2, M)** Stampa il grafico a barre verticali degli elementi di indice da **i1** a **i2** della sequenza, ossia rappresentando ciascuna barra con una sequenza verticale di caratteri **#**, e scalando la lunghezza delle barre in maniera tale che la barra corrispondente al valore massimo sia lunga **M** caratteri. Il valore massimo per (**i2 - i1**) è fissato a tempo di scrittura del programma. Non preoccuparsi di problemi di occupazione della memoria. Ad esempio, dato la sequenza di esempio al punto 2, se *i1=0*, *i2=3* ed **M=5**, stampa

```
#  
#  
##  
##  
## #
```

Se utile, una matrice dinamica di **n** righe e 10 colonne si può allocare e deallocare così:

```
<tipo_elem> (*p)[10] = new <tipo_elem>[n][10] ;      delete [] p;
```

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*. **Non preoccuparsi di problemi di precisione nel calcolo della lunghezza delle barre.**

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 (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
 - c) sono state seguite eventuali altre indicazioni presenti nella traccia in merito al voto finale