

# Programmazione I

## Prova di Programmazione – 30 gennaio 2013 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma che gestisce un insieme di pile di valori interi. Il numero di pile è noto a tempo di scrittura del programma ed all'avvio del programma le pile hanno lunghezza massima nulla (non vi si può quindi inserire nessun elemento). Realizzare le seguenti funzionalità.

1. **modifica\_capienza(N)** Cambia la lunghezza massima delle pile in N e le svuota. L'eventuale precedente contenuto delle pile è quindi perso.
2. **inserisci\_valore(v, i)** Inserisce il valore v nella pila di indice i.
3. **stampa\_insieme** Stampa la lunghezza massima delle pile ed il contenuto di ciascuna pila. Per ciascuna pila, stampa gli elementi in ordine inverso rispetto a quello in cui sono stati inseriti, ossia li stampa dall'elemento in cima a quello in fondo. Esempio:

**Capienza: 10**

**10 4 11 11**

**3 7**

**21 8 2 1 7**

4. **salva\_insieme** Salva l'insieme di pile in un file di testo dal nome predefinito.
5. **carica\_insieme** Carica l'insieme dal file. Il precedente contenuto dell'insieme è perso.
6. **stampa\_contenuto\_ordinato** Stampa l'indice ed il contenuto di ciascuna pila, in ordine di numero di elementi crescente. Ad esempio, partendo dall'esempio del punto 3, stamperebbe:

**1: 3 7**

**0: 10 4 11 11**

**2: 21 8 2 1 7**

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne gli *overflow* e l'inserimento di dati in formato errato da *stdin*.

---

### REGOLE

- Si può utilizzare ogni genere di manuale o di altro materiale didattico
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1, 2 e 3. 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) Nessuna funzionalità ha un costo computazionale più che lineare rispetto al numero di pile