# Prova di Laboratorio di Algoritmi
# 15 Giugno 2000, turno 1

Il testo è composto da quattro parti. Per superare l'esame è necessario superare almeno la prima parte. Le altre tre possono essere risolte in un ordine qualsiasi, essendo indipendenti le une dalle altre. Ad ogni parte deve corrispondere un solo file, contenente un programma scritto in linguaggio C *standard*. È ammesso utilizzare solamente le due librerie `stdio` e `stdlib`.

Si consiglia di leggere attentamente il testo in modo da seguirne correttamente tutte le direttive.

1. (file `parte1.c`, punti 15) Realizzare il tipo di dato successioni di interi $\geq 0$, utilizzando un'implementazione a puntatori.

   Il file `parte1.c` deve contenere le tre seguenti funzioni C:

   - `empty`: restituisce una successione vuota;
   - `insert`: prende un intero `i` $\geq 0$ ed una successione `s`, ed inserisce `i` **in coda** ad `s`;
   - `print`: stampa sullo standard output il contenuto di una successione, a partire dall'elemento **in testa**.

   La funzione `main` del file `parte1.c` deve realizzare (in ordine) i seguenti punti:

   - genera una successione vuota `s`, utilizzando la funzione `empty`;
   - inserimento nella successione `s` di una sequenza di interi $\geq 0$ letti da un file di input (usando la `insert` di cui sopra);
   - stampa (sullo standard output) di `s` (usando la `print` di cui sopra).

   **Il programma deve prevedere la lettura da tastiera del nome del file di input**. Il file di input è una sequenza (anche vuota) di interi $\geq 0$ (soliti separatori: blank, tabbing, new-line), terminata dal numero -1.

   **Esempio**:

   Se il file di input contiene i seguenti valori:

   ```
   45 32 80 0 78 65 8 20 98 67
   - 1
   ```

   allora l'output prodotto deve essere il seguente:

   ```
   45 32 80 0 78 65 8 20 98 67
   ```

2. (file `parte2.c`, punti 3) Estendere il file `parte1.c` in modo che contenga la funzione `num_occ` che prende un intero `i` $\geq 0$ ed una successione `s` e restituisce il numero di volte con cui `i` appare in `s`.

   Funzione `main`: come quella del file `parte1.c` più

   - lettura di un numero intero `i` $\geq 0$ da input;
   - stampa sullo standard output del numero di volte con cui `i` appare in `s`.

3. (file `parte3.c`, punti 5) Estendere il file `parte1.c` in modo che contenga la funzione `delete_occ` che prende una successione `s` ed un intero `i` e cancella da `s` tutti gli elementi $\geq$ `i`. La funzione **deve disallocare** lo spazio relativo alle celle cancellate.

   Funzione `main`: come quella del file parte1.c più

   - lettura da tastiera di numero intero `i`;
   - cancellazione da `s` di tutti gli elementi $\geq$ `i`, utilizzando la funzione `delete_occ`, sopra definita;
   - stampa della nuova successione, utilizzando la funzione `print`.

4. (file `parte4.c`, punti 7) Estendere il file `parte1.c` in modo che contenga la funzione `reverse` che prende una successione `s` e la inverte. La funzione **non deve allocare nuovo spazio di memoria**, ma utilizzare le celle di `s`.

   **Esempio**:

   Se la successione contiene i seguenti valori:

   ```
   45 32 80 0 78 65 8 20 98 67
   ```

   la funzione `reverse` deve restituire la seguente successione:

   ```
   67 98 20 8 65 78 0 80 32 45
   ```

   Funzione `main`: come quella del file parte1.c più

   - rovesciamento della successione `s`, utilizzando la funzione `reverse` sopra definita;
   - stampa della successione rovesciata, utilizzando la funzione `print`.