Prova di Laboratorio di Algoritmi
16 Giugno 2000, turno 3

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 insiemi di interi ≥ 0, utilizzando un'implementazione a puntatori.

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

   - `empty`: restituisce un insieme vuoto;
   - `insert`: prende un intero `i` ≥ 0 ed un insieme `s`, ed inserisce `i` in `s`, secondo l'**ordinamento crescente**;
   - `print`: stampa sullo standard output il contenuto di un insieme, a partire dall'elemento `in testa`.

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

   - genera un insieme vuoto `s`, utilizzando la funzione `empty`;
   - inserimento nell'insieme `s` di una sequenza di interi ≥ 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 ≥ 0 (soliti separatori: blank, tabbing, new-line), terminata dal numero -1.

   Esempio:

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

2. (file `parte2.c`, punti 3) Estendere il file `parte1.c` in modo che contenga la funzione `card` che prende un insieme `s` e ne restituisce la cardinalità.

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

   - stampa sullo standard output della cardinalità dell'insieme `s`.

3. (file `parte3.c`, punti 5) Estendere il file `parte1.c` in modo che contenga la funzione `subset` che prende in input due insiemi `s1` ed `s2` e restituisce vero se il primo insieme è contenuto nel secondo, falso altrimenti.

   Funzione `main`:

   - inserimento in `s1` e in `s2` di una sequenza di interi ≥ 0 letti da un file di input (usando la `insert` di cui sopra);
   - stampa (sullo standard output) di `s1` e `s2` (usando la `print` di cui sopra).
   - stampa di un opportuno messaggio sullo standard output in relazione all'esito della funzione `subset` applicata ad `s1` e `s2`.

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

   Esempio:

   ```
   45 32 80 0 78 65 8 20 98 67
   -1
   3 45 6 2 34
   -1
   ```

4. (file `parte4.c`, punti 7) Estendere il file `parte1.c` in modo che contenga la funzione `change_order` che prende una insieme `s` e genera l'insieme ordinato in senso decrescente. La funzione **non deve allocare nuovo spazio di memoria**, ma utilizzare le celle di `s`.

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

   - generazione insieme ordinato in senso decrescente, utilizzando la funzione `change_order` sopra definita;
   - stampa del nuovo insieme, utilizzando la funzione `print`.