Prova di Laboratorio di Algoritmi
8 luglio 1999, 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 insiemi di numeri reali $\neq 0$ utilizzando un'implementazione con **tavole hash aperte** contenenti un numero di buckets fornito in input dall'utente (quindi, nella struttura dati si dovrà memorizzare questo valore). Se `n` rappresenta il numero di buckets letti, la funzione di hash è definita come segue:

   ```
   int h(float i, int n)
   {
    int m;
    m = i;
    return m % n;
   }
   ```

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

   - `empty`: funzione che prende in input il numero di buckets `n` e restituisce una tavola vuota di dimensione `n`;
   - `insert`: inserisce un intero `i` nella tavola, **senza ripetizioni** e secondo l'ordinamento **crescente** rispetto al bucket nel quale l'elemento viene inserito.
   - `print`: stampa sullo standard output il contenuto di ogni bucket della tavola in input.

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

   - lettura del numero di buckets da input per la tavola `t`;
   - creazione di una tavola vuota `t` (usando la `empty` di cui sopra);
   - inserimento in `t` di una sequenza di interi letti da un file di input (usando la `insert` di cui sopra);
   - stampa (sullo standard output) di `t` (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 reali $\neq 0$ (soliti separatori: blank, tabbing, new-line), terminata dal numero 0.

Esempio:

Se `n = 5` e il file di input contiene:

```
8.65  12.75  4.1  34.0  67.02 23.9 1.0 15.2
0
```

allora l'output prodotto deve essere il seguente:

```
0: 15.2
1: 1.0
2: 12.75 67.02
3: 8.65  23.9
4: 4.1 34.0
```

2. (file `parte2.c`, punti 3) Estendere il file `parte1.c` in modo che contenga la funzione `min` che prende una tavola `t` e restituisce il valore minimo contenuto nella tavola. Se `n` identifica il numero di elementi contenuti nella tavola, la complessità di questa funzione deve essere **O(n)**, dove `n` identifica il numero dei buckets.

   Funzione `main`: come quella del file `parte1.c` più stampa dell'elemento minimo.

3. (file `parte3.c`, punti 5) Estendere il file `parte1.c` in modo che contenga la funzione `member` che prende un elemento `i` e una tavola `t` e restituisce vero se `i` è presente in `t`, falso altrimenti.

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

   - lettura di un intero `i` da input;
   - determinare se `i` è presente in `t`, usando la funzione `member` di cui sopra;
   - stampa di un opportuno messaggio.

4. (file `parte4.c`, punti 7) Estendere il file `parte1.c` in modo che contenga la funzione `union` che prende due tavole `t1` e `t2` ed aggiunge a `t1` tutti gli elementi di `t2`. A questo proposito, non devono essere allocate nuove celle di memoria, ma si devono utilizzare solo quelle già allocate in `t1` e `t2`. Si tenga presente che il numero di buckets in `t1` può essere diverso dal numero di buckets in `t2`, quindi lo stesso elemento nelle due tavole può essere contenuto in **buckets diversi**.

   Funzione `main`:

   - lettura del numero di buckets da input per la tavola `t1`;
   - lettura del numero di buckets da input per la tavola `t2`;
   - creazione delle tavole vuote `t1` e `t2` (usando la `empty` di cui sopra);
   - inserimento in `t1` e in `t2` di una sequenza di interi letti da un file di input (usando la `insert` di cui sopra);
   - stampa (sullo standard output) di `t1` e `t2` (usando la `print` di cui sopra).
   - costruzione della tavola insieme, utilizzando la funzione `union`;
   - stampa (sullo standard output, usando la `print`), della tavola unione;
   - stampa (sullo standard output, usando la `print`), delle tavole `t1` e `t2`.

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

Esempio:

```
8.65  12.75  4.1  34.0  67.02 23.9 1.0 15.2
0
3.25 8.01 3.98 34.2
0
```