



## ***Esercitazione di laboratorio n. 8***

(Caricamento sul portale entro le 23.59 del 12/12/2016 di entrambi gli esercizi)

### **Esercizio n. 1:** Questioni di equilibrio

Si consideri una sequenza di  $N$  frecce disposte lungo una linea orizzontale, in cui ogni freccia può puntare a destra oppure a sinistra, come illustrato nella figura seguente.



Si assuma che  $N$  sia sempre un numero pari e che la freccia più a sinistra occupi la posizione 0, mentre la freccia più a destra la posizione  $N-1$ .

Si definisce uno *scontro* quando due frecce consecutive puntano una verso l'altra. Nell'esempio precedente le frecce in posizione 0 e 1 e le frecce in posizione 4 e 5 determinano due scontri.

Dato uno scontro, la sequenza più lunga di frecce orientate nello stesso verso della freccia a sinistra (destra) dello scontro, rappresentano la parte sinistra (destra) dello scontro.

Uno scontro è in equilibrio se la parte destra e la parte sinistra dello scontro sono lunghe uguali. Nell'esempio di riferimento lo scontro (0,1) è equilibrato, mentre lo scontro (4,5) non lo è poiché la parte sinistra è lunga 3 mentre la parte destra è lunga 1.

Si scriva un programma in C in grado di determinare un insieme minimo di frecce da **girare** così che ogni freccia partecipi a uno scontro e ogni scontro sia equilibrato.

Come formato di input, si assuma che il valore 0 indichi una freccia orientata verso destra, mentre il valore 1 indichi una freccia orientata verso sinistra. La sequenza di 0 e di 1 è sempre preceduta dal numero di frecce prese in considerazione.

L'esempio precedente risulterebbe quindi codificato come segue: 6 0 1 0 0 0 1

In tale esempio, le condizioni imposte dal problema sarebbero rispettate scambiando di verso alla freccia in posizione 4.

### **Esercizio n.2:** Ricettario (rivisto)

Si consideri la medesima situazione introdotta nell'esercizio 3 del laboratorio 7. Si realizzi un programma C, organizzato su più moduli, che, attraverso un'apposita interfaccia utente, permetta di gestire una struttura dati adatta a memorizzare e gestire ricette.

Ogni ricetta sia caratterizzata da:

- il nome (stringa di massimo 255 caratteri, senza spazi)
- il tempo di preparazione, espresso in minuti
- l'elenco di ingredienti, ognuno in una certa quantità espressa in grammi.

Ogni ingrediente sia caratterizzato da:

- il nome (stringa di massimo 255 caratteri, senza spazi)
- il costo al chilo (numero reale positivo)
- l'apporto calorico al grammo (numero reale positivo).



Le informazioni relative agli ingredienti sono memorizzate in un file (`ingredienti.txt`), organizzato come segue:

- sulla prima riga è presente il numero `I` di ingredienti
- sulle `I` righe successive sono riportati i dettagli di ogni ingrediente, in ragione di una terna `<nome>` `<costo>` `<calorie>` per riga.

Le informazioni relative alle ricette sono memorizzate in un file (`ricette.txt`), organizzato come segue:

- sulla prima riga è presente il numero `R` di ricette
- seguono `R` blocchi di righe in cui sono riportati i dettagli di ogni ricetta. Ogni blocco ha il seguente formato:
  - sulla prima riga è presente una terna `<nome>` `<tempo>` `<num_ingredienti>`
  - seguono `num_ingredienti` righe, ognuna riportante una coppia `<nome_ingrediente>` `<grammi>`.

In questo esercizio, il programma deve essere realizzato su tre moduli distinti:

- `main` e interfaccia utente (il client)
- un modulo con le funzioni per la gestione degli ingredienti
- un modulo con le funzioni per la gestione delle ricette.

I singoli moduli devono offrire le funzionalità ritenute necessarie a gestire i rispettivi tipi, come caricamento in memoria di un dato, stampa di un dato, ecc.

Ai fini dell'esercizio, si introduca sia per gli ingredienti sia per le ricette una struttura *wrapper* per i due elenchi (le collezioni). La collezione di ricette va realizzata come lista concatenata (di ricette). Ogni ricetta, a sua volta, contiene una lista concatenata degli ingredienti contenuti (come riferimenti alla tabella degli ingredienti).

La tabella (collezione) degli ingredienti sia realizzata come vettore allocato dinamicamente.

Una volta acquisite le informazioni da file, le operazioni permesse dal programma devono essere quelle di:

- stampa dei dettagli di una ricetta
- stampa dei dettagli di un ingrediente
- calcolo del costo di una ricetta
- calcolo dell'apporto calorico di una ricetta
- stampa dell'intero elenco di ricette
- stampa dell'intero elenco di ingredienti
- stampa dell'elenco di ricette in cui sia presente un dato ingrediente
- introduzione (e memorizzazione) di una nuova ricetta (usando solo ingredienti già disponibili in elenco).

**Nota:** i contenuti del file di input di esempio in allegato sono indentati per rendere più facile distinguere una ricetta dalla successiva.