

## Prova scritta del 19/2/2015

*Convenzioni (obbligatorie): utilizzare i nomi indicati nel testo -  
 “indentare” in modo opportuno i programmi - inserire adeguati commenti*

1) Scrivere una funzione booleana di nome `permutazione` che, presi come suoi parametri due array di numeri interi `A` e `B`, il numero `n` di elementi presenti in `A` e in `B` (necessariamente lo stesso), restituisce `true` se gli elementi di `B` sono una permutazione degli elementi di `A`, `false` altrimenti. Ad esempio, se `A = {3, 5, 2, 8, 7}` e `B = {2, 3, 8, 5, 7}` il risultato della `permutazione` sarà `true`. Si assuma che `A` e `B` non contengano elementi ripetuti.

(b) Descrivere la funzione `permutazione` anche tramite diagramma di flusso.

(c) Realizzare un programma principale che legge da `std input` una sequenza di `n` numeri interi, terminata da un numero negativo, e li memorizza (per righe) in una matrice `M` di dimensioni `max 100 x 5` (si supponga che `n` sia un multiplo di 5). Il programma provvede quindi a stampare su `std output` la matrice `M`, ignorando tutte le righe di `M` che risultano permutazione della sua prima riga (utilizzare obbligatoriamente la funzione `permutazione`). Ad es., se `M = {{3, 5, 2, 8, 7}, {2, 3, 8, 5, 7}, {6, 5, 2, 9, 3}}`, la matrice stampata sarà:

```
3 5 2 8 7
6 5 2 9 3
```

2) Sia `coppia` il tipo di una struttura (`struct`) costituita da due campi, primo e secondo, entrambi di tipo `stringa` (dimensione `max. 50`).

(a) Realizzare una funzione booleana di nome `sostituisci` che, presi come suoi parametri un array `A` di elementi di tipo `coppia`, il numero `n` di elementi presenti in `A`, e due stringhe `s` e `r`, se `A` contiene un elemento con campo `primo` identico a `s`, restituisce in `r` la stringa contenuta nel campo `secondo` dell'elemento trovato e come suo risultato `true`; altrimenti, restituisce come risultato `false`.

(b) Scrivere un programma principale che: crea un array di nome `mappa` di elementi di tipo `coppia` inizializzato con `{{"per", "x"}, {"comunque", "cmq"}}`; legge da un file, il cui nome è fornito dall'utente tramite `std input` (lunghezza `max. 80` car.), una sequenza di stringhe, ciascuna terminata da un carattere “spazio”; per ciascuna stringa `s` letta, applica la funzione `sostituisci`, con `mappa` come primo e `s` come terzo parametro, e scrive su un file di output la stringa `r` determinata dalla `sostituisci`, o la stringa `s` stessa se la `sostituisci` restituisce `false`; quindi il programma termina. Il nome del file di output è ottenuto da quello di input aggiungendo la stringa “`_ridotto`”. Ad esempio:

*File di input* `testo`

`e' comunque vero per te e per me`

*File di output* `testo_ridotto`

`e' cmq vero x te e x me`

N.B. In entrambi gli esercizi utilizzare soltanto stringhe “*tipo C*”.