

Prova scritta del 14/6/2005

1) (a) Realizzare una funzione booleana di nome `ORDINATO` che, presi come suoi parametri un array di interi `A` ed il numero `n` di elementi in `A`, determina se `A` è ordinato in senso non-decrescente.

(b) Realizzare una funzione di nome `ELIMINA_ZERI` che, presi come suoi parametri un array di interi `V` ed il numero `n` di elementi in `V`, elimina da `V` tutti gli elementi il cui valore è zero. La funzione restituisce come suo risultato il numero di zeri eliminati da `V`.

(c) Descrivere la funzione `ORDINATO` anche tramite un diagramma di flusso.

2) Si consideri un file di numeri interi di nome `"numeri.dat"` contenente sequenze di numeri interi non-negativi separate tra loro da esattamente un numero negativo. Scrivere un programma che legge dal file `"numeri.dat"` le sequenze di numeri in esso contenute e, per ognuna di esse, determina, utilizzando opportunamente le due funzioni realizzate al punto (1), se la sequenza risulta ordinata in senso non-decrescente, a meno degli eventuali zero presenti in essa. Ad esempio, la sequenza `0 3 0 0 6 0 6 7 0` è ordinata, mentre la sequenza `3 0 0 1 0 3 4 5` non lo è. Il programma termina stampando su standard output il numero totale di sequenze ordinate trovate. Si supponga che la lunghezza max. di ciascuna sequenza sia `1000`.

SUGG.: Si memorizzi ciascuna sequenza di numeri letta da file in un array `S`, di capacità max. `1000`, che verrà quindi passato come parametro alle funzioni `ORDINATO` e `ELIMINA_ZERI` ...

3) Scrivere un programma principale che permetta di inizializzare e modificare una matrice `M` di dimensioni `10 x 20` i cui elementi sono strutture costituite da due campi, `C1` di tipo reale e `C2` di tipo booleano. Il campo `C2` indica se il corrispondente campo `C1` è inizializzato o no (e cioè contiene un valore significativo oppure no). Il programma propone all'utente un menù di possibili operazioni:

1. assegnare il valore `v` all'elemento di `M` di indici `i` e `j` (l'elemento individuato deve essere non-inizializzato e diventa inizializzato al termine dell'operazione);
2. sostituire il valore `v` di un elemento già inizializzato di indici `i` e `j` con un nuovo valore `w`;
3. eliminare il valore associato ad un elemento di indici `i` e `j` (l'elemento individuato deve essere inizializzato e diventa non-inizializzato al termine dell'operazione);
4. smettere.

Eseguita una delle prime tre operazioni, il programma ripropone il menù delle possibili operazioni finchè non viene scelta l'operazione 4, nel qual caso il programma termina definitivamente. `v`, `w`, `i` e `j` sono forniti dall'utente (su standard input) all'inizio dell'operazione scelta (si supponga, per semplicità, che i valori di `i` e `j` specificati siano sicuramente corretti). Se l'operazione non può essere eseguita a causa della inizializzazione/non-inizializzazione dell'elemento selezionato viene dato un opportuno messaggio d'errore ed il programma continua ripresentando il menù delle operazioni.

SUGG.: assegnare inizialmente il valore `false` ai campi `C2` di tutti gli elementi di `M`.