

Esercitazione del 18/01/2018

Esercizio 1. Si implementi in C++ una funzione che, ricevuti come parametri un array di caratteri **V** di lunghezza **n** ed un carattere **c**, restituisca *true* se nell'array **V** non capita mai che ci siano più di tre caratteri consecutivi diversi dal carattere **c**, *false* altrimenti.

Esempio:

Se l'array in input fosse $V = \{\mathbf{u}, \mathbf{a}, \mathbf{s}, \mathbf{f}, \mathbf{u}, \mathbf{d}, \mathbf{d}, \mathbf{u}, \mathbf{e}, \mathbf{r}, \mathbf{p}\}$ ed il carattere fosse $c = \mathbf{'u'}$, il metodo dovrebbe restituire *true*.
Se, invece, l'array in input fosse $V = \{\mathbf{x}, \mathbf{u}, \mathbf{y}, \mathbf{a}, \mathbf{s}, \mathbf{f}, \mathbf{u}, \mathbf{d}, \mathbf{d}\}$, ed il carattere sempre $c = \mathbf{'u'}$, il metodo dovrebbe restituire *false*: infatti è facile notare come si trovi almeno una sequenza composta da più di tre caratteri consecutivi diversi da **u** (in particolare, la sequenza : **y, a, s, f**).

Esercizio 2. Si implementi in C++ una funzione che, ricevuta come parametro una matrice di interi di dimensioni $N \times M$ contenente in tutte le celle il numero 0, legga da input una sequenza di numeri interi terminata dal numero "-1" e utilizzi tali numeri per popolare la matrice, seguendo un ordine "a serpentina". Ciò implica che la prima riga (indice 0) sarà popolata da sinistra verso destra, la seconda riga (indice 1) da destra verso sinistra, la terza riga (indice 2) da sinistra verso destra e così via...

Se la sequenza viene terminata prima di completare la matrice, allora le posizioni non utilizzate rimarranno impostate a 0. Se invece si leggono in sequenza $N \times M$ numeri diversi da -1, allora si deve comunque terminare la lettura producendo in output il messaggio: "Raggiunto il numero massimo di elementi!", senza proseguire oltre la lettura.

Esempio:

Supponendo che la matrice sia di dimensioni 4×3 e che la sequenza immessa sia: 1, 2, 3, 4, 5, 6, 7, -1 allora la matrice dovrebbe contenere i seguenti valori:

1	2	3
6	5	4
7	0	0
0	0	0

Supponendo, invece, che la sequenza immessa sia: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 allora la sequenza dovrebbe comunque essere terminata e dovrebbe essere prodotto in output il messaggio specificato precedentemente. Inoltre la matrice dovrebbe contenere i seguenti valori:

1	2	3
6	5	4
7	8	9
12	11	10

Esercizio 3. Scrivere in C++ un generatore di Sudoku semplificato. Il generatore deve riempire lo schema 9×9 con cifre casuali tra 1 e 9, e successivamente verificare che lo schema così riempito rispetti i requisiti del sudoku classico, ossia che ciascun numero compaia una ed una sola volta in ogni riga, colonna e settore.