

– Simulazione di Prova Intermedia –

12/01/2021

Questa è una simulazione di prova intermedia.

Nello svolgimento degli esercizi si invitano gli studenti a considerare aspetti di “buona programmazione”, “pulizia del codice” ed efficienza. Ad es.: formattazione corretta del codice, rendere il codice modulare aggiungendo ove necessario altri metodi rispetto a quelli richiesti dall'esercizio, soprattutto se questi rendono il codice più pulito e leggibile, o se evitano duplicazione di codice. Inoltre, non ci devono essere warning nel codice scritto.

Esercizio Java-- n. 1: Accoppiata

Un array di caratteri contiene una *accoppiata* se è composto da due sequenze di caratteri esattamente uguali. Ad esempio, l'array `{'m','a','t','r','i','x','m','a','t','r','i','x'}` contiene una accoppiata, mentre l'array `{'m','a','t','r','i','x'}` non la contiene.

Scrivere un metodo Java-- chiamato `accoppiata` che, dato in input un array `A` di caratteri composto da almeno un elemento, restituisca `true` se `A` contiene esattamente una accoppiata, e `false` altrimenti.

Nota bene: Gli studenti sono invitati a prediligere una soluzione ricorsiva. Una soluzione non ricorsiva (in cui si utilizzano istruzioni di iterazione, come cicli `for` o `while`) sarà comunque accettata e valutata dai docenti, ma la sua valutazione sarà penalizzata rispetto ad una soluzione ricorsiva.

Esercizio Java-- n. 2: Azzeramento Elementi Adiacenti

Sia T una matrice di numeri interi di dimensione $m \times n$ (m righe e n colonne), con $m \geq 3$ e $n \geq 3$, e siano r e c due numeri interi tali che $0 \leq r \leq m-1$ e $0 \leq c \leq n-1$.

L'azzeramento degli elementi adiacenti all'elemento di indice $[r,c]$ della matrice T , modifica la stessa matrice T ponendo uguale a 0 (quindi azzerando) tutti gli elementi adiacenti all'elemento di indice $[r,c]$, compresi quelli sulle diagonali. Nota: l'individuazione degli elementi adiacenti è da intendersi come *circolare*, ovvero:

- Gli elementi in riga $r=m-1$ (ultima riga della matrice) sono possibili adiacenti degli elementi in riga $r=0$ (prima riga della matrice), e viceversa;
- gli elementi in colonna $c=n-1$ (ultima colonna della matrice) sono possibili adiacenti degli elementi in colonna $c=0$ (prima colonna della matrice), e viceversa.

Ad esempio, sia $T = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$:

- Se $r=1$ e $c=1$, allora l'azzeramento degli elementi adiacenti all'elemento

$T[1,1]$ modifica la matrice T come segue: $T' = \begin{pmatrix} 0 & 0 & 0 & 4 \\ 0 & 6 & 0 & 8 \\ 0 & 0 & 0 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$.

- Se $r=0$ e $c=1$, allora l'azzeramento degli elementi adiacenti all'elemento

$T[0,1]$ modifica la matrice T come segue: $T' = \begin{pmatrix} 0 & 2 & 0 & 4 \\ 0 & 0 & 0 & 8 \\ 9 & 10 & 11 & 12 \\ 0 & 0 & 0 & 16 \end{pmatrix}$.

- Se $r=1$ e $c=0$, allora l'azzeramento degli elementi adiacenti all'elemento

$T[1,0]$ modifica la matrice T come segue: $T' = \begin{pmatrix} 0 & 0 & 3 & 0 \\ 5 & 0 & 7 & 0 \\ 0 & 0 & 11 & 0 \\ 13 & 14 & 15 & 16 \end{pmatrix}$.

- Se $r=3$ e $c=3$, allora l'azzeramento degli elementi adiacenti all'elemento

$T[3,3]$ modifica la matrice T come segue: $T' = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 5 & 6 & 7 & 8 \\ 0 & 10 & 0 & 0 \\ 0 & 14 & 0 & 16 \end{pmatrix}$.

Scrivere un metodo Java-- chiamato `azzeradiacenti` che, dato in input una matrice T di numeri interi di dimensione $m \times n$ (m righe e n colonne), con $m \geq 3$ e $n \geq 3$, e dati due numeri interi r e c tali che $0 \leq r \leq m-1$ e $0 \leq c \leq n-1$, restituisca la matrice T modificata applicando l'azzeramento degli elementi adiacenti all'elemento di indice $[r,c]$.