

Esercizio Java n. 1: Circolarità di una matrice

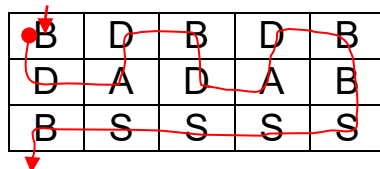
Esercizio estratto e adattato da Compito III Appello – 26/07/2023

Sia M una matrice di caratteri di dimensione $m \times n$, con $m > 0$ e $n > 0$, contenente i seguenti 4 possibili caratteri: 'A', 'B', 'S', 'D'. La matrice M è circolare se, partendo dalla cella di indici $(0,0)$, si ritorna nella stessa cella iniziale dopo aver visitato tutti gli elementi della matrice seguendo queste regole:

- Se si incontra il carattere 'A', la prossima cella da visitare è quella in "Alto" rispetto alla cella corrente;
- Se si incontra il carattere 'B', la prossima cella da visitare è quella in "Basso" rispetto alla cella corrente;
- Se si incontra il carattere 'S', la prossima cella da visitare è quella a "Sinistra" rispetto alla cella corrente;
- Se si incontra il carattere 'D', la prossima cella da visitare è quella a "Destra" rispetto alla cella corrente.

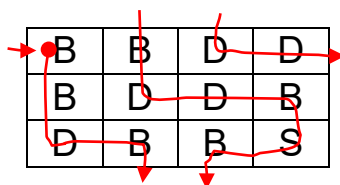
Nota bene: gli spostamenti sono da intendersi come circolari, ovvero uno spostamento verso destra a partire dall'ultima colonna porta alla prima colonna (e viceversa), e uno spostamento verso il basso dall'ultima riga porta alla prima riga (e viceversa).

Ad esempio, le seguenti matrici sono circolari (in **rosso** il cammino che riporta alla cella iniziale di indice $(0,0)$ dopo aver visitato tutte le celle della matrice):



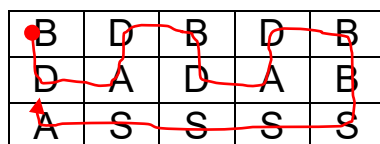
B	D	B	D	B
D	A	D	A	B
B	S	S	S	S

(esempio 1)



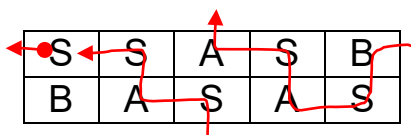
B	B	D	D
B	D	D	B
D	B	B	S

Ad esempio, le seguenti matrici non sono circolari:



B	D	B	D	B
D	A	D	A	B
A	S	S	S	S

← Si visitano tutte le celle ma non ritorno a quella iniziale.



S	S	A	S	B
B	A	S	A	S

← Ritorno alla cella iniziale ma non visito tutti gli elementi della matrice.

[CONSEGNA STANDARD]:

Scrivere un metodo Java-- di nome `circolare` che, dati in input una matrice M di caratteri come precedentemente descritto, restituisca `true` se la matrice è circolare, `false` altrimenti.

[CONSEGNA EXTRA - FACOLTATIVA]:

Si definisce “inversa” di una matrice M una matrice in cui tutti i caratteri ‘A’ vengono sostituiti con ‘B’ (e viceversa), e tutti i caratteri ‘S’ vengono sostituiti con ‘D’ (e viceversa). In pratica, si invertono tutte le direzioni alto-basso e sinistra-destra.

Ad esempio, l’inversa della matrice M dell’ *esempio 1* è la seguente matrice:

A	S	A	S	A
S	B	S	B	A
A	D	D	D	D

Scrivere un metodo Java-- di nome `circolareInversa` che, dati in input una matrice M di caratteri come precedentemente descritto, restituisca `true` se la matrice “inversa” di M è circolare, `false` altrimenti.

NOTA BENE:

- Saranno premiate le soluzioni che occuperanno meno memoria dati.
- I Junit Test da superare per la consegna standard sono quelli della classe **CircolareTest** (quelli della classe `CircolareInversaTest` falliranno).
- I Junit Test da superare per la consegna extra sono quelli della classe **CircolareInversaTest** (oltre a quelli della classe `CircolareTest` che devono comunque essere superati).
- Nello svolgere l’esercizio NON devono essere utilizzati i metodi `clone`, o `arraycopy`, o metodi della classe `Arrays`. L’utilizzo di tali metodi renderà l’esercizio automaticamente insufficiente.

Esercizio Java n. 2: Intero Bilanciato

Esercizio estratto e adattato da Compito IV Appello – 08/09/2023

Sia n un numero intero positivo composto da k cifre, con k pari e $k \geq 2$. Siano c_{k-1}, \dots, c_0 le k cifre di cui è composto n , ovvero $n = c_{k-1} \dots c_0$. Il numero n si dice “bilanciato” se valgono le seguenti condizioni:

$$c_{k-1} + c_0 == c_{k-2} + c_1 == \dots == c_{k/2} + c_{k/2-1} .$$

Ad esempio, il numero $n=61433250$ è bilanciato, poiché $6+0 == 1+5 == 4+2 == 3+3$.

Al contrario, il numero 234634 non è bilanciato, poiché $2+4 == 3+3 \neq 4+6$.

NOTA: un numero composto da due sole cifre è “bilanciato” solo se le due cifre di cui è composto sono uguali. Ad esempio: il numero $n=33$ è bilanciato, mentre il numero $n=35$ non è bilanciato.

Scrivere un metodo Java-- di nome `bilanciato` che, dati in input un numero intero positivo n composto da k cifre, con k pari e $k \geq 2$, restituisca `true` se il numero n è “bilanciato”, `false` altrimenti.

NOTA BENE:

- Saranno premiate le soluzioni che dichiareranno meno variabili di appoggio e occuperanno quindi meno memoria dati.
- E' possibile invocare il metodo `Math.pow` per calcolare la potenza di un numero.
- I Junit Test da superare sono quelli della classe **InteroBilanciatoTest**.
- Nello svolgere l'esercizio NON devono essere utilizzati i metodi `clone`, o `arraycopy`, o metodi della classe `Arrays`. L'utilizzo di tali metodi renderà l'esercizio automaticamente insufficiente.