Prova d'Esame del 25/02/2015

Esercizio 1. Si implementi in Java un metodo che, ricevuta come parametro una matrice di interi di dimensioni NxM 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 (quindi il metodo non le tocca). Se invece si leggono in sequenza NxM 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 4x3 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

SOLUZIONE:

```
import java.util.Scanner;
public class riempiMatriceASerpentina {
	public static Scanner input = new Scanner(System.in);

	public static void scriviMatrice(int m[][]) {
		for (int i = 0; i < m.length; i++) {
			for (int j = 0; j < m[i].length; j++)
				System.out.print(m[i][j] + " ");
			System.out.println();
		}
	}
	public static void riempiMatrice(int m[][]) {
		int contR = 0;
		int contC = 0;
		int cont = 0;
		System.out.println("Immetti una sequenza di numeri interi terminata da -1 ");
		int n = input.nextInt();
		while (n != -1 && cont < m.length * m[0].length){
			m[contR][contC]=n;
			cont++;
			if (contR % 2 == 0)
				if (contC == m[0].length-1)
					contR++;
				else
					contC++;
			else
				if (contC == 0)
					contR++;
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 25/02/2015

```
                        else
                                contC--;
                    if (cont < m.length * m[0].length)
                        n = input.nextInt();
                    else
                        System.out.println("Raggiunto il numero massimo di elementi che possono
essere contenuti nella matrice!");
                }
        }

        public static void main(String[] args) {
         System.out.println("Quante righe? ");
         int r = input.nextInt();
         System.out.println("Quante colonne? ");
         int c = input.nextInt();
         int m[][] = new int[r][c];
         riempiMatrice(m);
         scriviMatrice(m);
        }
}
```

Esercizio 2. Si implementi in Java un metodo che, ricevuti come parametri un array di caratteri (sia esso V) ed un carattere (sia esso c), restituisca *true* se nell'array V non capita mai che ci siano più di tre caratteri consecutivi diversi dal carattere c, *false* altrimenti.
N.B. Sarà riconosciuto un bonus se si implementa il metodo utilizzando la ricorsione.

Esempio:
Se l'array in input fosse V= {**u**, a, s, f, **u**, d, d, **u**, e, r, p} ed il carattere fosse c='u', il metodo dovrebbe restituire ***true***. Se, invece, l'array in input fosse V= { x, **u**, y, a, s, f, **u**, d, d}, ed il carattere sempre c='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).

SOLUZIONE:

```
import java.util.Scanner;
public class VerificaArrayCaratteri {
            public static Scanner input = new Scanner(System.in);

            public static void leggiArray(char a[]){
               System.out.println("Inserire " + a.length + " caratteri, elementi di un array");
               for (int i=0; i < a.length; i++)
                        a[i] = input.next().charAt(0);
            }

            public static boolean verificaArray(char v[], char c){
               int cont = 0;
               for (int i=0; i<v.length; i++){
                        if (v[i]==c)
                                cont = 0;
                        else{
                                cont++;
                                if (cont > 3)
                                        return false;
                        }
               }
               return true;
            }
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 25/02/2015

```
        public static boolean verificaArrayRicorsivo(char v[], char c, int pos, int cont){
           if (pos > v.length - 1)
                  return true;
           if (cont > 3)
                  return false;
           if (v[pos]==c)
                  cont = 0;
           else
                  cont++;
           return verificaArrayRicorsivo(v, c, pos+1, cont);
        }

        public static void main(String[] args) {
            System.out.println("Quanti elementi conterra' l'array? ");
            char v[] = new char[input.nextInt()];
            leggiArray(v);
            System.out.println("Inserire il carattere da verificare");
            char c = input.next().charAt(0);
            //if (verificaArray(v, c))
            if (verificaArrayRicorsivo(v, c, 0, 0))
                  System.out.println("ok");
            else
                  System.out.println("no");
        }
    }
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it