Code kata

Miscellanei

Nota bene: diversamente dagli altri Code Kata, in questi non vengono stabiliti con esattezza i dati, né l'output da produrre; entrambi dovranno essere definiti dallo studente, coerentemente con il metodo di lavoro mostrato in "Code Kata demo".

Indice generale

1 “Digital captcha”

Data una sequenza di cifre intere, calcolare la somma delle cifre corrispondenti a quelle successive nella sequenza. Nota bene: la sequenza è da intendersi “circolare”: la prima cifra rappresenta la cifra successiva all’ultima.

Esempi:

- `1122 -> 3`

  (la prima cifra corrisponde alla seconda; la terza alla quarta)

- `1111 -> 4`

  (ogni cifra corrisponde alla successiva)

- `1234 -> 0`

  (nessuna cifra corrisponde alla successiva)

- `91212129 -> 9`

  (l’ultima cifra corrisponde alla prima)

2 “Checksum”

Occorre calcolare il *checksum* di una griglia di valori numerici interi maggiori di zero.

Il procedimento è il seguente: per ogni riga della griglia si calcola la differenza tra il valore maggiore e quello minore. La somma delle differenze così calcolate da il *checksum*.

Ad esempio, la griglia:

5	1	9	5
7	5	3	4
2	4	6	8

produce i seguenti valori:

- 1ª riga: 9 – 1 → 8
- 2ª riga: 7 – 3 → 4
- 3ª riga: 8 – 2 → 6

Risultato finale: 18

Variazione

Considerare l’ipotesi che la griglia non sia rettangolare, e cioè che non tutte le righe abbiano la stessa lunghezza. Ad esempio:

5	1	9	5
7	5	3	
2	4	6	8

(Problema: in che modo rappresentare la griglia?)

3 “Passphrase”

Una *passphrase* valida (frase d’accesso) consiste in una serie di parole non ripetute. Ad esempio:

- `aa` `bb` `cc` `dd` → valida
- `aa` `bb` `cc` `dd` `aa` → non valida
- `aa` `bb` `cc` `dd` `aaa` → valida

Data una lista di *passphrase*, calcolare il numero di quelle valide.

(Problema: come rappresentare la singola *passphrase*?)

Variazione

Si suppone di memorizzare la singola *passphrase* in una stringa, separando le parole da uno spazio. Per ottenere le parole che compongono la frase occorre “dividere” la stringa mediante il metodo `Split()`:

```
string frase = "aa bb cc dd";
string[] parola = frase.Split();
//-> aa, bb, cc, dd
```

4 Funzione 1° grado

Calcolare le "y" della funzione di 1° grado: **y = x * 2**, con la **x** che varia da 0 a 10, con un passo di 0.5. Restituire un vettore di `List<double>` contenente le y.

```
//-> 0, 1, 2, 5, 6, ...
```

Variazioni

1. Generalizzare il procedimento, parametrizzando gli estremi dell'intervallo e il passo di variazione della **x**.

   ```
   double minX = -10;
   double maxX = 10;
   double deltaX = 5;
   //-> -20, -10, 0, 10, 20
   ```

2. Restituire una lista di record contenente le coppie (x, y).