

Appello TAP del 02/07/2013

Scrivere nome, cognome e matricola sul foglio protocollo, indicando anche se avete nel piano di studi TAP da 8 CFU (quello attuale) o da 6 CFU (quello “vecchio”).

Chi deve sostenere TAP da 6 CFU non dovrà svolgere un esercizio (indicato nel seguito); per loro il punteggio indicato nel testo sarà scalato, di conseguenza.

Esercizio 1 (10 punti)

- Implementare, usando NUnit, una serie di test per verificare che il metodo di sistema `Enumerable.Single<T>(IEnumerable<T>, Func<T, boolean>)` rispetti la sua specifica (trovate una copia della documentazione MSDN nel foglio allegato allo scritto).
- Dare, se possibile, un’implementazione *sbagliata* del metodo che passi tutti i vostri test oppure, spiegare perché non è possibile.

Esercizio 2 (5 punti) — solo per chi ha TAP da 8 CFU

- Dare l’implementazione del metodo asincrono `SumLengthsAsync` che, preso in input un array di `Task<string>`, restituisca un `Task<int>` che corrisponde a sommare le lunghezze di tutte le stringhe prodotte dai task dell’array.
- Dare un esempio di invocazione del metodo implementato nel punto precedente.

Esercizio 3 (15 punti)

Scrivere l’extension-method generico `RL<T>` che, presa una sequenza, anche infinita, s (di elementi di tipo T) e un intero non negativo n , restituisce la sequenza s “ruotata a sinistra” di n posizioni. Ovvero, se $s = e_1, \dots, e_k$, $s.RL(n) = e_{n+1}, e_{n+2}, \dots, e_k, e_1, \dots, e_n$.

Nei casi in cui s è vuota, il risultato dovrà essere la sequenza vuota; negli altri casi in cui $n > |s|$, il risultato dovrà essere equivalente a ruotare s di $(n \bmod |s|)$ posizioni, dove con $|s|$ abbiamo indicato la lunghezza della sequenza e con \bmod il resto della divisione intera.

Sollevare delle eccezioni, standard o definite da voi, ove ritenuto necessario.

Esempio di uso (ricordando che le stringhe sono `IEnumerable<char>`):

```

private static void TestRL(string s, int n) {
    Console.WriteLine("\{0}\".RL(\{1})=\{2}\", s, n,
                      new string(s.RL(n).ToArray()));
}

static IEnumerable<int> Evens() {
    for (int n = 0;; n += 2)
        yield return n;
}

private static void Main(string[] args) {
    const string hw = "Hello world!";
    TestRL(hw, 0);
    TestRL(hw, 3);
    TestRL(hw, 12);
    TestRL(hw, 14);
    TestRL(string.Empty, 0);
    TestRL(string.Empty, 1);
    TestRL(string.Empty, 10000);
    const string a = "a";
    TestRL(a, 0);
    TestRL(a, 1);
    TestRL(a, 10);
    var x = Evens().RL(7).GetEnumerator();
    for (int i = 0; i < 5; ++i) {
        var b = x.MoveNext();
        Debug.Assert(b);
        Console.WriteLine(x.Current);
    }
}

```

che dovrebbe visualizzare:

```

"Hello world!".RL(0)="Hello world!"
"Hello world!".RL(3)="lo world!Hel"
"Hello world!".RL(12)="Hello world!"
"Hello world!".RL(14)="llo world!He"
"" .RL(0)=""
"" .RL(1)=""
"" .RL(10000)=""
"a" .RL(0)="a"
"a" .RL(1)="a"
"a" .RL(10)="a"
14
16
18
20
22

```