Appello TAP del 25/06/2015

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") e se avete consegnato i test durante l'anno e intendete usufruire del bonus così conseguito.

Chi deve sostenere TAP da 6 CFU non deve svolgere l'esercizio 1 e chi ha consegnato i test durante l'anno (e intende usufruire del bonus conseguito) non deve svolgere l'esercizio 4; per loro il punteggio indicato nel testo sarà scalato, di conseguenza, in modo che il massimo conseguibile sia sempre 30. Avete a disposizione mezzora per esercizio. In sintesi:

Tipo TAP	Bonus Test	Esercizi da svolgere	Tempo a disposizione	Fattore di normalizzazione
6CFU	sì	2 e 3	1h	$\frac{30}{10+9}$
6CFU	no	2, 3 e 4	1h 30'	$\frac{30}{10+9+6}$
8CFU	sì	1, 2 e 3	1h 30'	$\frac{30}{5+10+9}$
8CFU	no	tutti (1, 2, 3 e 4)	2h	1

Esercizio 1 (punti 5) Si confrontino i seguenti metodi, spiegandone le differenze in uno o più casi d'uso.

```
public Task<int> TaskMaxAsync1(Task<int>[] my_ints)
{
    return Task.WhenAll(my_ints)
        .ContinueWith(x => x.Result.Where(i => i%2 != 0).Max());
}

public Task<int> TaskMaxAsync2(Task<int>[] my_ints)
{
    var numbers = Task.WhenAll(my_ints).Result;
    return Task.FromResult(numbers.Where(i => i%2 != 0).Max());
}
```

Esercizio 2 (punti 10) Si utilizzi la seguente funzione

```
public static string ToMorse(this Char inChar) { ... }
```

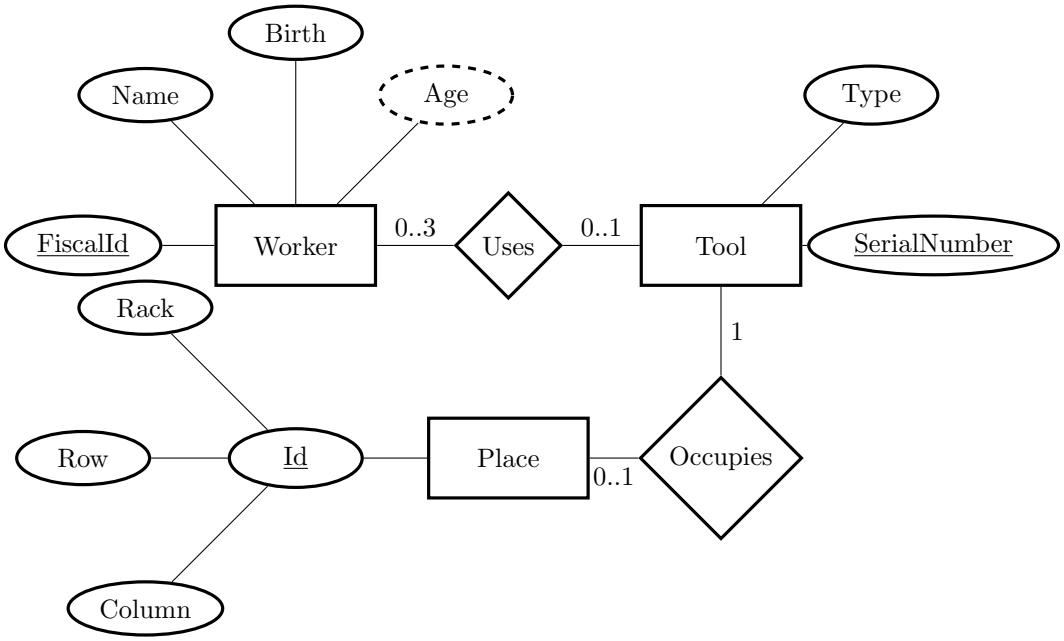
che traduce i caratteri supportati nel corrispondente codice *morse* e restituisce la stringa vuota su ogni altro carattere, per scrivere l'extension-method `CodingToMorse` che, presa una sequenza (potenzialmente infinita) di caratteri, restituisce una sequenza (potenzialmente infinita) di caratteri (linee e punti) che la traducono in alfabeto morse.

Il metodo dovrà prendere come parametro ("this") `message`, la sequenza da tradurre, di tipo `IEnumerable<char>` e sollevare l'eccezione...

- `ArgumentNullException` se `message` è null;
- `ArgumentException` se `message` contiene un carattere intraducibile, ovvero su cui `ToMorse` restituisce la stringa vuota.

Esercizio 3 (punti 9) Si implementi una componente per gestire una base di dati che memorizzi gli operai di un'officina e gli attrezzi a loro disposizione, come descritto dal seguente diagramma E/R.

Ciascuno attrezzo ha una collocazione riservata in cui deve essere riposto quando a disposizione. Ciascun operaio può usare contemporaneamente al più tre attrezzi. Come ulteriore vincolo del dominio si sa che tutti i `Rack` hanno 20 righe, ciascuna delle quali ha 40 colonne.



Nel diagramma, gli attributi sottolineati sono chiavi e quelli in ovali tratteggiati sono derivati.

Le operazioni sui dati, *di cui è richiesta* ***solo*** *l'intestazione* (non l'implementazione) e che sono elencate per permettere una corretta modellazione dei dati, sono le seguenti:

- inserimento di un nuovo lavoratore;
- eliminazione di un lavoratore; eventuali attrezzi assegnatigli torneranno a disposizione;
- inserimento di un nuovo attrezzo, con assegnazione automatica della posizione fra quelle disponibili;
- eliminazione di un attrezzo, possibile solo se non è al momento in uso;
- assegnazione di un attrezzo (al momento libero) ad un lavoratore (che non ne abbia già in uso il massimo consentito);
- rilascio di un attrezzo da parte di un lavoratore a cui non serve più, che lo ripone al suo posto;
- inserimento dei nuovi posti, a seguito del posizionamento di un nuovo scaffale;
- ricerca degli attrezzi correntemente assegnati ad un lavoratore.

Esercizio 4 (punti 2+2+2 = 6 punti)

- Elencare, descrivendoli a parole, una lista di test significativi per il metodo `CodingToMorse`, dell'esercizio 2.
- Implementare, usando NUnit, due test della lista precedente; uno che vada a testare un caso "buono" (ovvero, dove ci si aspetta che l'invocazione di `CodingToMorse` vada a buon fine) e uno che vada a testare un caso "cattivo" (ovvero, dove ci si aspetta che l'invocazione di `CodingToMorse` sollevi un'eccezione).
- Implementare, usando NUnit, un test in cui `CodingToMorse` venga invocato su un argomento "infinito".