

Esercitazione C# .NET

Esercizio 1

- Scrivere un programma in C# .NET Rubrica.exe
- Leggere la prima riga del file rubrica.json
- Il file contiene una rubrica telefonica
- Ogni riga del file è un JSON nel seguente formato:

```
{ "Nome": "Mario", "Cognome": "Rossi", "Numero": "3312345678" }
```
- Utilizzare System.Text.Json
- Fare riferimento agli esempi su github.com/corso-bdx se necessario
- Aiutarsi con Google (o altri strumenti esterni) se necessario, query di ricerca: «Come leggere un JSON in C# con System.Text.Json»
- Stampare su console il primo contatto nel seguente formato:
Mario Rossi: 3312345678

Esercizio 2

- Leggere tutto il file rubrica.json
- Stampare su console tutti i contatti, uno per riga, nello stesso formato dell'esercizio precedente

Esempio:

```
PS C:\Users\User> Rubrica.exe
```

```
Mario Rossi: 3312345678
```

```
Filippo Russo: 3361943025
```

```
Gabriel Ferrari: 3366942053
```

```
Stefano Esposito: 3297309974
```

```
Thomas Bianchi: 3284220560
```

```
Edoardo Colombo: 3454192010
```

```
Gioele Romano: 3221433417
```

```
Alice Ricci: 3105523827
```

```
Gabriele Gallo: 3331601351
```

```
Noemi Dal: 3283427119
```

Esercizio 3

- Leggere il primo parametro `args[0]` da linea di comando
- Se è `"lista"`, stampare la lista come da esercizio precedente
- Se è `"cerca"`, usare Linq per stampare solo i contatti che presentano `args[1]` nel nome, cognome o numero

Esempio 1:

```
PS C:\Users\User> Rubrica.exe lista
```

```
Mario Rossi: 3312345678  
Filippo Russo: 3361943025  
Gabriel Ferrari: 3366942053  
Stefano Esposito: 3297309974  
Thomas Bianchi: 3284220560  
Edoardo Colombo: 3454192010  
Gioele Romano: 3221433417  
Alice Ricci: 3105523827
```

Esempio 2:

```
PS C:\Users\User> Rubrica.exe cerca Maria
```

```
Carlotta Mariani: 3109574027  
Maria Cattaneo: 3110907717
```

Esempio 3:

```
PS C:\Users\User> Rubrica.exe cerca 3326
```

```
Nicole Grasso: 3326400505
```

Esercizio 4

- Se `args[0]` non è stato indicato, usare `Console.ReadLine()` per chiedere quale operazione eseguire

Esempio:

```
PS C:\Users\User> Rubrica.exe
```

Operazioni disponibili: lista, cerca

Cosa vuoi fare? lista

Mario Rossi: 3312345678

Filippo Russo: 3361943025

Gabriel Ferrari: 3366942053

Stefano Esposito: 3297309974

Thomas Bianchi: 3284220560

Edoardo Colombo: 3454192010

Gioele Romano: 3221433417

Alice Ricci: 3105523827

Gabriele Gallo: 3331601351

Noemi Dal: 3283427119

Esercizio 5

- Aggiungere comando "nuovo" per aggiungere una voce al file rubrica.json
- Usare `Console.ReadLine()` per chiedere nome, cognome e numero

Esempio:

```
PS C:\Users\User> Rubrica.exe nuovo
```

```
Nome: Renato
```

```
Cognome: Verdi
```

```
Numero: 3394826712
```

```
Contatto aggiunto!
```

Esercizio 6

- Quando si usa il comando "**nuovo**", se sono presenti `args[1]`, `args[2]` ed `args[3]`, usarli come nome, cognome e numero
- Leggere i parametri mancanti con `Console.ReadLine()`

Esempio 1:

```
PS C:\Users\User> Rubrica.exe nuovo  
Nome: Renato  
Cognome: Verdi  
Numero: 3394826712  
Contatto aggiunto!
```

Esempio 2:

```
PS C:\Users\User> Rubrica.exe nuovo Renato Verdi 3394826712  
Contatto aggiunto!
```

Esempio 3:

```
PS C:\Users\User> Rubrica.exe nuovo Renato  
Cognome: Verdi  
Numero: 3394826712  
Contatto aggiunto!
```

Esercizio 7

- Aggiungere comando "cancella" per rimuovere una voce dal file rubrica.json
- Se non è indicato args[0], usare Console.ReadLine() per chiedere il nome o cognome o numero del contatto da cancellare

Esempio 1:

```
PS C:\Users\User> Rubrica.exe cancella  
Chi vuoi cancellare? Mario  
Cancellato Mario Rossi (3312345678)
```

Esempio 2:

```
PS C:\Users\User> Rubrica.exe cancella  
Chi vuoi cancellare? Gerardo  
Gerardo non trovato.
```

Esempio 3:

```
PS C:\Users\User> Rubrica.exe cancella Rossi  
Cancellato Mario Rossi (3312345678)
```

Esempio 4:

```
PS C:\Users\User> Rubrica.exe cancella Paolo  
Nome ambiguo:  
Paolo Giuliani: 3409923195  
Paolo Rota: 3365512149  
PS C:\Users\User> Rubrica.exe cancella 3409923195  
Cancellato Paolo Giuliani (3409923195)
```


Esercizio 8

- Separare la logica dei comandi dal codice che interpreta il comando
- Creare un nuovo progetto di tipo "Libreria di classi" per contenere la logica dei vari comandi

Esercizio 9

- Creare un nuovo progetto di tipo "App Windows Form"
- Creare un'interfaccia grafica per interagire con la rubrica
- Usare la libreria di classi creata nell'esercizio precedente, adattarla se necessario

Esercizio Bonus 1

- Aggiungere comando **"esporta"** per generare una pagina HTML contenente i contatti il rubrica

Esempio:

PS C:\Users\User> Rubrica.exe esporta
Esportato file rubrica.html

```
rubrica.html
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <title>Rubrica</title>
</head>
<body>
  <ul>
    <li>Mario Rossi: <a href="tel:3312345678">3312345678</a></li>
    <li>Filippo Russo: <a href="tel:3361943025">3361943025</a></li>
    <li>Gabriel Ferrari: <a href="tel:3366942053">3366942053</a></li>
    <li>Stefano Esposito: <a href="tel:3297309974">3297309974</a></li>
    <li>Thomas Bianchi: <a href="tel:3284220560">3284220560</a></li>
    <!-- ecc... -->
  </ul>
</body>
</html>
```

Esercizio Bonus 2

- Aggiungere parametro **"html"** al comando **"esporta"** per esportare in formato HTML come da esercizio precedente
- Aggiungere parametro **"vcard"** al comando **"esporta"** per esportare in formato vCard

Esempio 1:

```
PS C:\Users\User> Rubrica.exe esporta html  
Esportato file rubrica.html
```

Esempio 2:

```
PS C:\Users\User> Rubrica.exe esporta vcard  
Esportato file rubrica.vcf
```

Esempio 3:

```
PS C:\Users\User> Rubrica.exe esporta  
Esportato file rubrica.html
```

```
rubrica.vcf  
BEGIN:VCARD  
VERSION:3.0  
FN:Mario Rossi  
N:Rossi;Mario;;;  
TEL;TYPE=CELL:3312345678  
END:VCARD  
BEGIN:VCARD  
VERSION:3.0  
FN:Filippo Russo  
N:Russo;Filippo;;;  
TEL;TYPE=CELL:3361943025  
END:VCARD  
BEGIN:VCARD  
VERSION:3.0  
FN:Gabriel Ferrari  
N:Ferrari;Gabriel;;;  
TEL;TYPE=CELL:3366942053  
END:VCARD
```

The background is a solid blue color with a repeating pattern of small, white, stylized icons. These icons represent various concepts related to technology, business, and communication, such as a smartphone, a laptop, a cloud, a lightbulb, a gear, a network diagram, a document, a bar chart, a pie chart, a speech bubble, a magnifying glass, a rocket, a handshake, a lightbulb, a gear, a network diagram, a document, a bar chart, a pie chart, a speech bubble, a magnifying glass, a rocket, a handshake, and others.

Buon divertimento!