



Corso di Laurea in Informatica

TESI DI LAUREA

Titolo della tesi di laurea

**Relatore:**

**Nome Cognome**

**Candidato:**

**Nome Cognome**

**Correlatore:**

**Nome Cognome**

**ANNO ACCADEMICO 20xx/20xx**

# Indice

<b>Elenco delle figure</b>	<b>ii</b>
<b>Elenco dei codici sorgente</b>	<b>iii</b>
<b>Introduzione</b>	<b>1</b>
<b>1 Hello World 1</b>	<b>2</b>
1.1 S1 - Hello World . . . . .	4
1.1.1 Sottosezione 1 . . . . .	4
<b>2 Capitolo 2 di test</b>	<b>5</b>
2.0.1 Sottosezione 1 . . . . .	5
<b>Conclusioni</b>	<b>8</b>

# Elenco delle figure

1.1	Esempio di un'immagine . . . . .	2
-----	----------------------------------	---

# Elenco dei codici sorgente

1.1	Codice JavaScript . . . . .	3
2.1	Codice C . . . . .	6
2.2	Codice Java . . . . .	8

# Introduzione

Introduzione...

# 1. Hello World 1

Capitolo 1...

**Hello World!** Today I am learning  $\text{\LaTeX}$ .  $\text{\LaTeX}$  is a great program for writing math. I can write in line math such as  $a^2 + b^2 = c^2$ . I can also give equations their own space:

$$\gamma^2 + \theta^2 = \omega^2 \tag{1.1}$$

If I do not leave any blank lines  $\text{\LaTeX}$  will continue this text without making it into a new paragraph. Notice how there was no indentation in the text after equation (1). Also notice how even though I hit enter after that sentence and here ↓  $\text{\LaTeX}$  formats the sentence without any break. Also look how it doesn't matter how many spaces I put between my words.

For a new paragraph I can leave a blank space in my code.

Esempio di una nota<sup>1</sup> con citazione a sezione 1.1.



Figura 1.1: Esempio di un'immagine

---

<sup>1</sup>CleanCode

```

1  function foo( array )
2  {
3      let sume = 0;
4      let counte = 0;
5      let sumo = 0;
6      let counto = 0;
7      for ( var n of array )
8          {
9              var even = n % 2;
10             ( even ) == 0 ? sume += n : sumo += n;
11             ( even ) == 0 ? counte++ : counto++;
12         }
13     var re = counte == 0 ? 0 : sume/counte;
14     var ro = counto == 0 ? 0 : sumo/counto;
15     return [re, ro];
16 }
17
18
19 const array = new Array();
20 let stop = false
21
22 do
23     {
24         let v = prompt( "insert: " );
25         if( v != "STOP!" )
26             {
27                 v = Number(v);
28                 array.push( v );
29             }
30         else
31             {
32                 stop = true;
33             }
34     }
35 while( !stop );

```

Code 1.1: Codice JavaScript

## 1.1 S1 - Hello World

Sezione 1... con citazione bibliografica [?]

### 1.1.1 Sottosezione 1

Sottosezione 1...

$$\bar{X} \pm \frac{\sqrt{\hat{p}(1-\hat{p})}}{\sqrt{n}} q_{1-\frac{\alpha}{2}} = \hat{p} \pm \frac{\sqrt{\hat{p}(1-\hat{p})}}{\sqrt{n}} q_{1-\frac{\alpha}{2}}$$



## 2. Capitolo 2 di test

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse ac lectus dolor. Nunc id ligula vitae odio pellentesque semper. Vestibulum commodo, nisi a consequat pellentesque, nisi libero semper lorem, in tincidunt nibh sapien vel felis. Duis tincidunt sapien mi, in eleifend urna pulvinar ac. Quisque porta at nunc nec auctor. Etiam molestie massa ante, non euismod nisi aliquam id. Nullam eu quam ac nulla mattis vestibulum quis in ligula. Etiam faucibus magna erat, at pulvinar ligula lobortis id. Etiam felis elit, congue in enim vitae, congue ultricies diam. Suspendisse lacus nibh, pretium sit amet fermentum nec, dapibus sodales dolor. Sed neque enim, tincidunt ac rutrum at, congue in sapien. Cras sodales turpis est, vitae luctus est mollis quis.

$$A = \begin{bmatrix} A_{n-1} & z \\ w^T & a_{nn} \end{bmatrix} = L \cdot U = \begin{bmatrix} L_{n-1} & 0 \\ x^T & 1 \end{bmatrix} \cdot \begin{bmatrix} U_{n-1} & y \\ 0 & w_{nn} \end{bmatrix} =$$

### 2.0.1 Sottosezione 1

Vestibulum auctor odio nec fermentum eleifend. In mollis laoreet dolor, et tempus nisi pharetra eu. Donec non lorem sed mi ullamcorper hendrerit. Sed pellentesque aliquam semper. Phasellus in elit erat. Curabitur consectetur vulputate metus sed fermentum. Nullam lobortis purus nec dui molestie imperdiet a at quam. Etiam eget pharetra erat, quis dignissim quam. Vivamus quam purus, tincidunt id urna et, mattis bibendum nunc. Nulla fermentum scelerisque quam pellentesque aliquam. Nunc lacinia rutrum orci, vitae venenatis ipsum congue quis. Aliquam in est est.

```

1  void destroy_queue( queue_t *q )
2  {
3      if( q != NULL )
4      {
5          clear_queue( q );
6
7          pthread_mutex_lock( &q->queue_lock );
8          if( q->head->ptr != NULL )
9              free( q->head->ptr );
10         free( q->head );
11         pthread_mutex_unlock( &q->queue_lock );
12
13         free( q );
14     }
15 }

```

Code 2.1: Codice C

Suspendisse imperdiet fringilla ornare. Sed ultrices eros ex, in consectetur massa elementum eu. Nam quis tristique mauris. Ut gravida dapibus nisl, id tempus metus porttitor vel. Pellentesque viverra metus nec ante pharetra dictum. Duis tincidunt sem vel ex placerat placerat. Donec nec orci vitae arcu placerat suscipit pretium at massa. Aliquam suscipit faucibus orci sed mattis. Curabitur et consectetur nibh. In hac habitasse platea dictumst. Nullam molestie diam nec arcu tempus interdum. Quisque pulvinar ut ligula posuere consequat. Ut ac pharetra lectus, in condimentum nisi.

Vestibulum vestibulum commodo tellus et tristique. Sed fermentum, purus eu tempor hendrerit, risus purus varius nibh, a laoreet metus neque pharetra lorem. Donec non accumsan arcu. Integer accumsan mollis erat, eget convallis dui eleifend ut. Aenean luctus purus a augue semper, at pulvinar nulla congue. Vestibulum risus arcu, varius eget rhoncus sed, venenatis vitae nisl. Morbi lacinia, nisl a tincidunt fringilla, lacus metus accumsan arcu, in convallis risus magna in tellus. Ut gravida ultrices dictum. Maecenas placerat mi eu nulla ullamcorper, at

rhoncus risus condimentum.

In eu ultrices nibh. Aenean egestas consectetur semper. Pellentesque ornare pulvinar ullamcorper. Donec mattis erat a aliquet porta. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nulla porttitor hendrerit sem, vel malesuada libero cursus sed. Duis mollis fermentum elit sit amet tincidunt. Donec dictum nisl ac mi consequat commodo. Vivamus at sollicitudin elit. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin ac lorem quis tortor euismod sagittis in nec diam. Mauris faucibus erat in nunc blandit interdum et eu mi.

# Conclusioni

**Teorema 1** (Teorema fondamentale calcolo integrale). Sia  $f(x)$  una funzione continua e limitata in  $[a, b[$ . La sua funzione integrale

$$F(x) = \int_a^x f(t) dt$$

è derivabile, e si ha  $F'(x) = f(x)$ .

Se poi  $G(x)$  è una funzione derivabile e  $G'(x) = f(x)$  allora si ha  $F(x) = G(x) - G(a)$

```
1 public class Test {
2     public static void main (String[] args) {
3         SMap<Integer, String> test = new TwoListSMap<Integer, String>();
4
5         test.put(new Integer (1) , "uno");
6         test.put(new Integer (30), "trenta");
7
8         test.get(new Integer (30) );
9
10        System.out.println( test.getKeys().toString() );
11        System.out.println( test.toString() );
12    }
13 }
```

Code 2.2: Codice Java

# Bibliografia

- [1] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*.  
Prentice Hall PTR, USA, 1 edition, 2008.