



C Piscine

C 01

*Summary: Questo documento è il subject del Modulo C 01 della C Piscine in 42*

*Version: 6*

# Contents

I	Istruzioni	2
II	Istruzioni sull'AI	4
III	Preludio	7
IV	Esercizio 00 : ft_ft	8
V	Esercizio 01 : ft_ultimate_ft	9
VI	Esercizio 02 : ft_swap	10
VII	Esercizio 03 : ft_div_mod	11
VIII	Esercizio 04 : ft_ultimate_div_mod	12
IX	Esercizio 05 : ft_putstr	13
X	Esercizio 06 : ft_strlen	14
XI	Esercizio 07 : ft_rev_int_tab	15
XII	Esercizio 08 : ft_sort_int_tab	16
XIII	Consegna e valutazione tra pari	17

# Chapter I

## Istruzioni

- Dovrai usare solo questo documento come riferimento, non fidarti delle voci di corridoio.
- Assicurati di avere i permessi appropriati su file e cartelle.
- Ricordati di seguire le **procedure di consegna**.
- I tuoi esercizi saranno controllati e valutati dai tuoi compagni.
- Inoltre, i tuoi esercizi saranno valutati da un programma chiamato **Moulinette**.
- **Moulinette** è meticolosa e severa nella valutazione. È un programma e non è possibile negoziare con lei. Per evitare brutte sorprese, sii il più scrupoloso possibile.
- **Moulinette** non è di larghe vedute. Se il tuo codice non rispetta la Norma, non cercherà di capirlo. **Moulinette** si affida a un programma chiamato **norminette** per verificare che i tuoi file rispettino la Norma. TL;DR: Consegnare un progetto che non supera il controllo di **norminette** non ha senso.
- Questi esercizi sono ordinati per difficoltà, dal più facile al più difficile. **Non** verrà considerato un esercizio più difficile completato con successo se uno più facile non è completamente funzionante.
- Usare una funzione proibita è considerato cheating. I cheater ricevono un voto di **-42**, non negoziabile.
- Devi consegnare una funzione **main()** solo se viene specificatamente richiesto un **programma**.
- **Moulinette** compila con le seguenti opzioni: **-Wall -Wextra -Werror**, usando **cc**.
- Se il tuo programma non compila, riceverai un voto di **0**.
- **Non puoi** lasciare **alcun** file aggiuntivo nella tua cartella oltre a quelli specificati nel **subject**.
- Hai una domanda? Chiedi al compagno alla tua destra. Se no, prova con quello alla tua sinistra.

- La tua guida di riferimento si chiama **Google / man / Internet / ...**
- Esamina attentamente gli esempi. Potrebbero contenere dettagli cruciali non esplicitamente indicati nel subject...



Non dimenticare di aggiungere l'*header standard 42* in ciascuno dei tuoi file .c/.h. Norminette ne verifica comunque l'esistenza!



Norminette deve essere lanciata con il flag *-R CheckForbiddenSourceHeader*. Anche Moulinette lo userà.

# Chapter II

## Istruzioni sull'AI

### ● Contesto

La Piscine di C è intensa. È la tua prima grande sfida in 42 — un'immersione profonda nel problem-solving, nell'autonomia e nella comunità.

In questa fase, il tuo obiettivo principale è costruire le basi — attraverso la fatica, la ripetizione e soprattutto il **peer-learning**.

Nell'era dell'AI, le scorciatoie sono facili da trovare. Tuttavia, è importante chiedersi se l'uso dell'AI ti stia davvero aiutando a crescere — o se stia semplicemente ostacolando lo sviluppo di competenze reali.

La Piscine è anche un'esperienza umana — e per ora, nulla può sostituirla. Nemmeno l'AI.

Per una panoramica più completa della nostra posizione sull'AI — come strumento di apprendimento, come parte del curriculum ICT e come aspettativa crescente nel mercato del lavoro — consulta la FAQ dedicata disponibile sull'intranet.

### ● Messaggio principale

- 👉 Costruisci solide basi senza scorciatoie.
- 👉 Sviluppa davvero competenze tecniche e trasversali.
- 👉 Vivi il vero peer-learning, inizia a imparare come imparare e risolvere nuovi problemi.
- 👉 Il percorso di apprendimento è più importante del risultato.
- 👉 Impara i rischi associati all'AI e sviluppa pratiche di controllo efficaci e contromisure per evitare errori comuni.

## ● **Regole per chi apprende:**

- Dovresti applicare il ragionamento ai problemi assegnati, soprattutto prima di rivolgerti all'AI.
- Non dovresti chiedere risposte dirette all'AI.
- Dovresti informarti sull'approccio globale di 42 riguardo all'AI.

## ● **Risultati della fase:**

In questa fase di base, otterrai i seguenti risultati:

- Acquisirai solide basi tecniche e di programmazione.
- Capirai perché e come l'AI può essere pericolosa in questa fase.

## ● **Commenti ed esempi:**

- Sì, sappiamo che l'AI esiste — e sì, può risolvere i tuoi progetti. Ma sei qui per imparare, non per dimostrare che l'AI ha imparato. Non perdere tempo (né il nostro) solo per dimostrare che l'AI può risolvere il problema dato.
- Imparare a 42 non significa conoscere la risposta — significa sviluppare la capacità di trovarne una. L'AI ti dà la risposta direttamente, ma così ti impedisce di costruire il tuo ragionamento. E il ragionamento richiede tempo, impegno e comporta fallimenti. Il percorso verso il successo non deve essere facile.
- Ricorda che durante gli esami, l'AI non è disponibile — niente internet, niente smartphone, ecc. Ti renderai subito conto se hai fatto troppo affidamento sull'AI durante il tuo percorso di apprendimento.
- Il peer-learning ti espone a idee e approcci diversi, migliorando le tue competenze interpersonali e la capacità di pensare in modo divergente. Questo è molto più prezioso che parlare con un bot. Quindi non essere timido — parla, fai domande e impara insieme agli altri!
- Sì, l'AI farà parte del curriculum — sia come strumento di apprendimento che come argomento a sé. Avrai anche la possibilità di creare il tuo software di AI. Per saperne di più sul nostro approccio graduale, consulta la documentazione disponibile sull'intranet.

### ✓ **Buona pratica:**

Sono bloccato su un nuovo concetto. Chiedo a qualcuno vicino come lo ha affrontato. Parliamo per 10 minuti — e all'improvviso capisco. Ci sono riuscito.

**✗ Bad practice:**

Uso di nascosto l'AI, copio del codice che sembra giusto. Durante la peer evaluation non riesco a spiegare nulla. Fallisco. Durante l'esame — senza AI — sono di nuovo bloccato. Fallisco.

# Chapter III

## Preludio

Vincent: E sai come chiamano un... un... un Quarter Pounder with Cheese a Parigi?

Jules: Non lo chiamano Quarter Pounder with cheese?

Vincent: No amico, hanno il sistema metrico. Non saprebbero cos'è un Quarter Pounder.

Jules: E allora come lo chiamano?

Vincent: Lo chiamano Royale with cheese.

Jules: Una Royale with cheese. E come chiamano un Big Mac?

Vincent: Beh, un Big Mac è un Big Mac, ma lo chiamano "Le Big Mac".

Jules: "Le Big-Mac." Ah ah ah ah. E come chiamano un Whopper?

Vincent: Non so, non sono andato al Burger King.

Almeno uno dei seguenti esercizi non ha nulla a che vedere con una Royale with cheese.

# Chapter IV

## Esercizio 00 : ft\_ft

	Exercise 00
	ft_ft
Directory:	<i>ex00/</i>
Files to Submit:	<b>ft_ft.c</b>
Authorized:	None

- Crea una funzione che prende come parametro un puntatore a un int e imposta il valore di quell'int a "42".
- La funzione deve essere prototipata come segue:

```
void ft_ft(int *nbr);
```

# Chapter V

## Esercizio 01 : ft\_ultimate\_ft

	Exercise 01
	ft_ultimate_ft
Directory:	<i>ex01/</i>
Files to Submit:	<b>ft_ultimate_ft.c</b>
Authorized:	None

- Crea una funzione che prende come parametro un puntatore a un int e imposta il valore di quell'int a “42”.
- La funzione deve essere prototipata come segue:

```
void      ft_ultimate_ft(int *****nbr);
```

# Chapter VI

## Esercizio 02 : ft\_swap

	Exercise 02
	ft_swap
Directory:	<i>ex02/</i>
Files to Submit:	<b>ft_swap.c</b>
Authorized:	None

- Crea una funzione che scambia i valori di due interi usando i loro indirizzi ricevuti come parametri.
- La funzione deve essere prototipata come segue:

```
void ft_swap(int *a, int *b);
```

# Chapter VII

## Esercizio 03 : **ft\_div\_mod**

	Exercise 03
	ft_div_mod
Directory:	<i>ex03/</i>
Files to Submit:	<b>ft_div_mod.c</b>
Authorized:	None

- Crea una funzione **ft\_div\_mod** con il seguente prototipo:

```
void      ft_div_mod(int a, int b, int *div, int *mod);
```

- Questa funzione divide ‘a’ per ‘b’ e memorizza il risultato nell’intero puntato da ‘div’. Memorizza anche il resto della divisione di ‘a’ per ‘b’ nell’intero puntato da ‘mod’.

# Chapter VIII

## Esercizio 04 : **ft\_ultimate\_div\_mod**

	Exercise 04
	<b>ft_ultimate_div_mod</b>
Directory:	<i>ex04/</i>
Files to Submit:	<b>ft_ultimate_div_mod.c</b>
Authorized:	None

- Crea una funzione **ft\_ultimate\_div\_mod** con il seguente prototipo:

```
void      ft_ultimate_div_mod(int *a, int *b);
```

- Questa funzione divide il valore puntato da ‘a’ per il valore puntato da ‘b’. Il risultato della divisione viene memorizzato nell’intero puntato da ‘a’, mentre il resto viene memorizzato nell’intero puntato da ‘b’.

## Soglia di oggi

La soglia di validazione per questo progetto è 50%.

Sta a te determinare quali esercizi ti permettono di raggiungere questa soglia, e se vuoi completarne altri.

# Chapter IX

## Esercizio 05 : ft\_putstr

	Exercise 05
	ft_putstr
Directory:	<i>ex05/</i>
Files to Submit:	<b>ft_putstr.c</b>
Authorized:	<b>write</b>

- Crea una funzione che stampa una stringa di caratteri sull'output standard.
- La funzione deve essere prototipata come segue:

```
void      ft_putstr(char *str);
```

# Chapter X

## Esercizio 06 : ft\_strlen

	Exercise 06
	ft_strlen
Directory:	<i>ex06/</i>
Files to Submit:	<b>ft_strlen.c</b>
Authorized:	None

- Crea una funzione che conta e restituisce il numero di caratteri in una stringa.
- La funzione deve essere prototipata come segue:

```
int      ft_strlen(char *str);
```

# Chapter XI

## Esercizio 07 : ft\_rev\_int\_tab

	Exercise 07
	ft_rev_int_tab
Directory:	<i>ex07/</i>
Files to Submit:	<b>ft_rev_int_tab.c</b>
Authorized:	None

- Crea una funzione che inverte un array di interi dato (il primo elemento diventa l'ultimo, e così via).
- La funzione prende due argomenti: un puntatore a un int e il numero di elementi dell'array.
- La funzione deve essere prototipata come segue:

```
void    ft_rev_int_tab(int *tab, int size);
```

# Chapter XII

## Esercizio 08 : ft\_sort\_int\_tab

	Exercise 08
	ft_sort_int_tab
Directory:	<i>ex08/</i>
Files to Submit:	<b>ft_sort_int_tab.c</b>
Authorized:	None

- Crea una funzione che ordina un array di interi in ordine crescente.
- La funzione prende due argomenti: un puntatore a un int e il numero di elementi dell'array.
- La funzione deve essere prototipata come segue:

```
void    ft_sort_int_tab(int *tab, int size);
```

# Chapter XIII

## Consegna e valutazione tra pari

Consegna il tuo progetto nel tuo repository **Git** come al solito. Solo il progetto all'interno del tuo repository sarà valutato durante la difesa. Assicurati di ricontrillare i nomi dei tuoi file per essere sicuro che siano corretti.



Devi restituire solo i file richiesti dal soggetto di questo progetto.