



C Piscine

C 03

Sommario: Questo documento è il soggetto per il modulo C 03 della C Piscine @ 42.

Indice

I	Istruzioni	2
II	Istruzioni sull'AI	4
III	Prefazione	7
IV	Esercizio 00 : ft_strcmp	8
V	Esercizio 01 : ft_strncmp	9
VI	Esercizio 02 : ft_strcat	10
VII	Esercizio 03 : ft_strncat	11
VIII	Esercizio 04 : ft_strstr	12
IX	Esercizio 05 : ft_strlcat	13
X	Consegna e valutazione tra pari	14

Capitolo 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...



Norminette sarà lanciato con il flag `-R CheckForbiddenSourceHeader`.
Anche Moulinette lo userà.

Capitolo 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.

Capitolo III

Prefazione

La prima menzione conosciuta del gioco Carta-Forbice-Sasso (RPS) appare nel libro *Wuzazu*, scritto dall'autore cinese della dinastia Ming Xie Zhaozhi. Egli notò che il gioco risale alla dinastia Han cinese (206 a.C. – 220 d.C.). Nel *Wuzazu*, il gioco era chiamato *shoushiling*.

Anche il libro di Li Rihua *Note of Liuyanzhai* fa riferimento a questo gioco, chiamandolo *shoushiling*, *huozhitou* o *huoquan*.

Durante la storia giapponese, ci sono frequenti menzioni di *sansukumi-ken*, che significa giochi “ken” (pugno) con un *sukumi* (stallo) a tre vie. In questi giochi, A batte B, B batte C e C batte A. I giochi sono nati in Cina prima di essere introdotti in Giappone, dove sono diventati molto popolari.

All'inizio del XX secolo, Carta-Forbice-Sasso si era diffuso oltre l'Asia, soprattutto grazie all'aumento delle interazioni giapponesi con l'Occidente. Il suo nome inglese deriva da una traduzione dei tre gesti giapponesi che rappresentano sasso, carta e forbici. Altrove in Asia, il gesto del palmo aperto rappresenta tipicamente il “tessuto” piuttosto che la “carta”. Anche la rappresentazione delle forbici segue lo stile giapponese.

Nel 1927, *La Vie au Patronage*, una rivista francese per bambini, descrisse il gioco in dettaglio, riferendosi ad esso come *jeu japonais* (“gioco giapponese”). Il suo nome francese, *Chi-fou-mi*, si basa sulle antiche parole giapponesi per “uno, due, tre” (*hi, fu, mi*).

Un articolo del 1932 del *New York Times* sull'ora di punta di Tokyo spiegava le regole del gioco ai lettori americani, suggerendo che non fosse ancora molto conosciuto negli Stati Uniti all'epoca. L'edizione del 1933 di *Compton's Pictured Encyclopedia* lo descriveva come un metodo comune per i bambini giapponesi per risolvere le dispute, chiamandolo “John Kem Po”. L'articolo notava anche: “Questo è un modo così efficace per decidere una discussione che anche i ragazzi e le ragazze americani potrebbero volerlo praticare.”

Capitolo IV

Esercizio 00 : ft_strcmp

	Esercizio 00
	ft_strcmp
	Cartella per la consegna : <i>ex00/</i>
	File da consegnare : ft_strcmp.c
	Funzioni permesse : Nessuna

- Riproduci il comportamento della funzione `strcmp` (man `strcmp`).
- La funzione deve essere prototipata come segue:

```
int      ft_strcmp(char *s1, char *s2);
```

Capitolo V

Esercizio 01 : ft_strcmp

	Esercizio 01
	ft_strcmp
	Cartella per la consegna : <i>ex01/</i>
	File da consegnare : ft_strcmp.c
	Funzioni permesse : Nessuna

- Riproduci il comportamento della funzione **strcmp** (man strcmp).
- La funzione deve essere prototipata come segue:

```
int      ft_strcmp(char *s1, char *s2, unsigned int n);
```

Capitolo VI

Esercizio 02 : ft_strcat

	Esercizio 02
	ft_strcat
	Cartella per la consegna : <i>ex02/</i>
	File da consegnare : ft_strcat.c
	Funzioni permesse : Nessuna

- Riproduci il comportamento della funzione **strcat** (man **strcat**).
- La funzione deve essere prototipata come segue:

```
char *ft_strcat(char *dest, char *src);
```

Capitolo VII

Esercizio 03 : ft_strncat

	Esercizio 03
	ft_strncat
	Cartella per la consegna : <i>ex03/</i>
	File da consegnare : ft_strncat.c
	Funzioni permesse : Nessuna

- Riproduci il comportamento della funzione **strncat** (man strncat).
- La funzione deve essere prototipata come segue:

```
char *ft_strncat(char *dest, char *src, unsigned int nb);
```

Capitolo VIII

Esercizio 04 : ft__strstr

	Esercizio 04
	ft__strstr
	Cartella per la consegna : <i>ex04/</i>
	File da consegnare : ft__strstr.c
	Funzioni permesse : Nessuna

- Riproduci il comportamento della funzione **strstr** (man strstr).
- La funzione deve essere prototipata come segue:

```
char *ft__strstr(char *str, char *to_find);
```

Capitolo IX

Esercizio 05 : ft_strlcat

	Esercizio 05
	ft_strlcat
	Cartella per la consegna : <i>ex05/</i>
	File da consegnare : ft_strlcat.c
	Funzioni permesse : Nessuna

- Riproduci il comportamento della funzione **strlcat** (man strlcat).
- La funzione deve essere prototipata come segue:

```
unsigned int ft_strlcat(char *dest, char *src, unsigned int size);
```

Capitolo X

Consegna e valutazione tra pari

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



Devi consegnare solo i file esplicitamente richiesti dai requisiti del progetto.