

C Piscine

C 00

Sommario: Questo documento serve come subject per il modulo C 00 della C Piscine presso 42.

Indice

I	Istruzioni	2
II	Istruzioni sull'AI	4
III	Prefazione	7
IV	Esercizio 00: ft_putchar	8
V	Esercizio 01: ft_print_alphabet	9
VI	Esercizio 02: ft_print_reverse_alphabet	10
VII	Esercizio 03: ft_print_numbers	11
VIII	Esercizio 04: ft_is_negative	12
IX	Esercizio 05: ft_print_comb	14
X	Esercizio 06: ft_print_comb2	15
XI	Esercizio 07: ft_putnbr	16
XII	Esercizio 08: ft_print_combn	17
XIII	Consegna e valutazione tra peer	18

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...



Non dimenticare di includere l'*header standard 42* in ciascuno dei tuoi file `.c` e `.h`. Norminette ne controllerà comunque la presenza!



Norminette deve essere eseguito 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

L'olio di fegato di merluzzo è un integratore alimentare derivato dal fegato del merluzzo (Gadidae).

Come la maggior parte degli oli di pesce, contiene alti livelli di acidi grassi omega-3, inclusi acido eicosapentaenoico (EPA) e acido docosaesaenoico (DHA). L'olio di fegato di merluzzo è anche ricco di vitamine A e D.


Storicamente, è stato consumato per il suo contenuto di vitamina A e vitamina D.

Un tempo veniva comunemente dato ai bambini, poiché è stato dimostrato che la vitamina D previene il rachitismo e altri sintomi di carenza di vitamina D.

Al contrario dell'olio di fegato di merluzzo, il C è buono, assicurati di mangiarne un po'!

Capitolo IV

Esercizio 00: ft_putchar

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

- Scrivi una funzione che visualizza il carattere passato come parametro.
- La funzione deve essere prototipata come segue:


```
void ft_putchar(char c);
```

- Per visualizzare il carattere, devi usare la funzione **write** come segue:

```
write(1, &c, 1);
```

Capitolo V

Esercizio 01: ft_print_alphabet

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

- Crea una funzione che visualizza l'alfabeto in minuscolo, su una sola riga, in ordine crescente, a partire dalla lettera 'a'.
- La funzione deve essere prototipata come segue:


```
void ft_print_alphabet(void);
```



Non esitare a chiedere a qualcuno nel tuo cluster se hai una domanda.

Capitolo VI

Esercizio 02: ft_print_reverse_alphabet

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

- Crea una funzione che visualizza l'alfabeto in minuscolo, su una sola riga, in ordine decrescente, a partire dalla lettera 'z'.
- La funzione deve essere prototipata come segue:


```
void ft_print_reverse_alphabet(void);
```



Fai il push del tuo codice su Git regolarmente!

Capitolo VII

Esercizio 03: ft_print_numbers

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

- Crea una funzione che visualizza tutte le cifre su una sola riga, in ordine crescente.
- La funzione deve essere prototipata come segue:


```
void ft_print_numbers(void);
```



La collaborazione è la chiave del successo.

Capitolo VIII

Esercizio 04: ft_is_negative

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

- Crea una funzione che visualizza 'N' o 'P' a seconda del segno dell'intero passato come parametro.
 - Se **n** è negativo, visualizza 'N'.
 - Se **n** è positivo o zero, visualizza 'P'.
- La funzione deve essere prototipata come segue:

```
void ft_is_negative(int n);
```



Il fallimento fa parte del tuo percorso di apprendimento.

Traguardo raggiunto, continua così!

Hai raggiunto la fine degli esercizi obbligatori richiesti per validare questo progetto.


Ora sta a te decidere se continuare con i seguenti esercizi opzionali o passare al prossimo progetto. Entrambi i percorsi ti esporranno a concetti preziosi.

Per prendere la tua decisione, considera i seguenti punti:

- Il primissimo esame si concentra sulla programmazione in C. Se hai già lavorato al primo progetto C, questa esperienza sarà utile. Lo stesso vale per la "rush" alla fine della settimana (presto ne saprai di più).
- Il tuo successo in questa Piscine viene valutato in base a diversi fattori. Completare ogni progetto è importante, ma anche il tuo progresso complessivo su tutti i progetti della Piscine gioca un ruolo chiave. Scegli saggiamente per ottimizzare i tuoi risultati.
- Puoi sempre rivedere e riprovare lo stesso progetto tra qualche giorno o settimana, fino alla fine della Piscine.
- Rimanere al passo con i tuoi peer favorisce una migliore collaborazione.

Capitolo IX

Esercizio 05: ft_print_comb

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

- Crea una funzione che stampa tutte le combinazioni uniche di tre cifre distinte, con sia le cifre all'interno di ogni combinazione sia le combinazioni stesse in ordine crescente.

- Output atteso:

```
$>./a.out | cat -e  
012, 013, 014, 015, 016, 017, 018, 019, 023, ..., 789$>
```

- **987** non è inclusa perché **789** copre già quella combinazione.
- **999** non è inclusa perché la cifra **9** appare più di una volta.
- La funzione deve essere prototipata come segue:


```
void ft_print_comb(void);
```



Hai controllato con il tuo vicino a destra?

Capitolo X

Esercizio 06: ft_print_comb2

	Esercizio 06
	ft_print_comb2
	Cartella per la consegna : <i>ex06/</i>
	File da consegnare : ft_print_comb2.c
	Funzioni permesse : write

- Crea una funzione che stampa tutte le diverse combinazioni di due numeri distinti di due cifre (XX XX) tra **00** e **99**, elencate in ordine crescente.

- Output atteso:

```
$>./a.out | cat -e
00 01, 00 02, 00 03, 00 04, 00 05, ..., 00 99, 01 02, ..., 97 99, 98 99$>
```

- La funzione deve essere prototipata come segue:


```
void ft_print_comb2(void);
```



Ispirati agli altri, ma non lasciare che facciano il lavoro per te!

Capitolo XI

Esercizio 07: ft_putnbr

	Esercizio 07
	ft_putnbr
	Cartella per la consegna : <i>ex07/</i>
	File da consegnare : ft_putnbr.c
	Funzioni permesse : write

- Crea una funzione che stampa il numero passato come parametro. La funzione deve essere in grado di stampare tutti i valori possibili di una variabile di tipo **int**.
- La funzione deve essere prototipata come segue:

```
void ft_putnbr(int nb);
```


- Esempio:
 - **ft_putnbr(42);** deve visualizzare **42**.



Non fidarti di una sola fonte di informazioni, esegui sempre i tuoi test, controlli e verifiche!

Capitolo XII

Esercizio 08: ft_print_combn

	Esercizio 08
	ft_print_combn
	Cartella per la consegna : <i>ex08/</i>
	File da consegnare : ft_print_combn.c
	Funzioni permesse : write

- Crea una funzione che stampa tutte le combinazioni uniche di **n** cifre distinte in ordine crescente, senza ripetizione.
- Il valore di **n** sarà tale che: $0 < n < 10$.
- Esempio di output per **n = 2**:

```
$> ./a.out | cat -e  
01, 02, 03, ..., 09, 12, ..., 79, 89$>
```

- La funzione deve essere prototipata come segue:

```
void ft_print_combn(int n);
```



Hai controllato con il tuo vicino a sinistra?

Capitolo XIII

Consegna e valutazione tra peer

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



Devi inviare solo i file specificati nelle istruzioni del progetto.