



Piscina C

C 09

Sommario: Questo documento tratta il modulo C 09 della Piscina C @ 42.

Indice

I	Istruzioni	2
II	Preambolo	4
III	Esercizio 00 : libft	5
IV	Esercizio 01 : Makefile	6
V	Esercizio 02 : ft_split	8

Capitolo I

Istruzioni

- Fate riferimento solo a questa pagina: non fidatevi delle dicerie.
- Questo documento può subire variazioni prima della scadenza per la presentazione.
- Controllate i permessi dei vostri file e delle vostre cartelle.
- Dovete seguire le procedure di presentazione per tutti gli esercizi.
- I vostri esercizi saranno controllati e valutati dai vostri compagni di corso.
- Moulinette sarà estremamente meticolosa e severa nel valutare il vostro lavoro. Essendo il suo un processo automatico senza possibilità di ricorso, assicuratevi di essere il più precisi possibile al fine di evitare brutte sorprese.
- I vostri esercizi saranno soggetti, oltre alla valutazione tra pari, al controllo e alla valutazione da parte di un programma chiamato Moulinette.
- Moulinette non ha una mentalità aperta. Non proverà a comprendere il vostro codice se non rispetta la Norma. Moulinette utilizza un programma di nome **norminette** per controllare la validità dei vostri file. TL;DR: sarebbe scocco tentare di consegnare un esercizio che non pass il controllo di **norminette**.
- Gli esercizi sono presentati seguendo un ordine di difficoltà crescente. Ai fini della valutazione **NON** si prendono in considerazione gli esercizi se i precedenti non sono stati completati correttamente
- Usare una funzione non autorizzata viene considerato come barare. Chi bara ottiene un **-42** senza possibilità di ricorso.
- Dovrete consegnare una funzione `main()` solo se l'esercizio richiede un programma.
- Moulinette compila per mezzo di `gcc` utilizzando queste flag: `-Wall -Wextra -Werror`.
- Se il vostro programma non compila, il voto sarà 0.
- NON sarà tollerato ALCUN file aggiuntivo nelle cartelle presentate oltre a quelli specificati in questo documento.

- Dubbi o domande? Chiedi a chi si trova alla tua destra, altrimenti a chi si trova alla tua sinistra
- Your reference guide is called `Google / man / the Internet /`
- Date un'occhiata alla sezione Piscina C del forum dell'Intranet.
- Prestate attenzione agli esempi proposti, in quanto potrebbero mostrare dettagli non esplicitamente presentati nel documento...
- Per Odin, Per Thor ! Usate la testa !!!



Norminette va utilizzata con la flag `-R CheckForbiddenSourceHeader`.
Moulinette farà la stessa cosa.

Capitolo II

Preambolo

Dialog from the movie The Big Lebowski:

The Dude: Walter, ya know, it's Smokey, so his toe slipped over the line a little, big deal. It's just a game, man.

Walter Sobchak: Dude, this is a league game, this determines who enters the next round robin. Am I wrong? Am I wrong?

Smokey: Yeah, but I wasn't over. Gimme the marker Dude, I'm marking it 8.

Walter Sobchak: [pulls out a gun] Smokey, my friend, you are entering a world of pain.

The Dude: Walter...

Walter Sobchak: You mark that frame an 8, and you're entering a world of pain.

Smokey: I'm not...

Walter Sobchak: A world of pain.

Smokey: Dude, he's your partner...

Walter Sobchak: [shouting] Has the whole world gone crazy? Am I the only one around here who gives a shit about the rules? Mark it zero!

The Dude: They're calling the cops, put the piece away.

Walter Sobchak: Mark it zero!

[points gun in Smokey's face]

The Dude: Walter...


Walter Sobchak: [shouting] You think I'm fucking around here? Mark it zero!

Smokey: All right, it's fucking zero. Are you happy, you crazy fuck?

Walter Sobchak: ...It's a league game, Smokey.

Capitolo III

Esercizio 00 : libft

	Esercizio 00
	libft
	Cartella per la consegna : <i>ex00/</i>
	File da consegnare : <i>libft_creator.sh</i> , <i>ft_putchar.c</i> , <i>ft_swap.c</i> , <i>ft_putstr.c</i> , <i>ft_strlen.c</i> , <i>ft_strcmp.c</i>
	Funzioni permesse : <i>write</i>

- Creare la libreria *libft.a*.
- Uno script di shell di nome *libft_creator.sh* dovrà occuparsi di compilare e creare la libreria.
- La libreria deve contenere tutte le seguenti funzioni :

```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

- Sarà utilizzato il seguente comando :

```
sh libft_creator.sh
```

Capitolo IV

Esercizio 01 : Makefile

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

- Create un **Makefile** che compili la libreria **libft.a**.
- Deve stampare tutti i comandi che sta eseguendo.
- Non deve eseguire nessun comando non necessario.
- Il **Makefile** cercherà i source file nella cartella "srcs".
- I source file saranno: `ft_putchar.c`, `ft_swap.c`, `ft_putstr.c`, `ft_strlen.c`, `ft_strcmp.c`
- Il **Makefile** cercherà i file di header nella cartella "includes".
- Quei file saranno: `ft.h`
- Deve compilare i file `.c` per mezzo di `gcc` utilizzando le flag `-Wall` `-Wextra` `-Werror`.
- La libreria dovrà essere al root dell'esercizio..
- I file `.o` dovranno essere insieme ai file `.c`.
- Il **Makefile** deve anche implementare le seguenti regole: **clean**, **fclean**, **re**, **all** e ovviamente **libft.a**.
- Eseguire **make** dovrebbe avere lo stesso effetto di eseguire **make all**
- La regola **all** deve avere lo stesso effetto di eseguire **make libft.a**.
- La regola **clean** deve rimuovere tutti i file temporanei generati.
- La regola **fclean** deve tenere lo stesso comportamento **make clean** ed in più rimuovere tutti i file binari creati con **make all**.

- La regola `re` deve equivalere all'esecuzione di un `make fclean` seguito da un `make all`.
- Il vostro Makefile non deve compilare file inutilmente.
- Utilizzeremo il vostro Makefile con i nostri file.



Attenzione alle Wildcards!

Capitolo V

Esercizio 02 : ft_split

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

- Creare una funzione che divida una stringa di caratteri sulla base di una seconda stringa di caratteri.
- Dovrete usare i caratteri presenti nella stringa **charset** come separatori
- La funzione restituirà un array dove ogni elemento conterrà l'indirizzo di una stringa compresa tra due separatori. L'ultimo elemento dovrà essere uguale a 0 per indicare la fine dell'array.
- L'array non può contenere stringhe vuote. Traete le vostre conclusioni.
- La stringa passata come argomento non sarà modificabile.
- Il prototipo è il seguente :

```
char **ft_split(char *str, char *charset);
```