



Piscina C

C 11

Sommario: Questo documento tratta il modulo C 11 della Piscina C @ 42.

Indice

I	Istruzioni	2
II	Preambolo	4
III	Esercizio 00 : ft_foreach	6
IV	Esercizio 01 : ft_map	7
V	Esercizio 02 : ft_any	8
VI	Esercizio 03 : ft_count_if	9
VII	Esercizio 04 : ft_is_sort	10
VIII	Esercizio 05 : do-op	11
IX	Esercizio 06 : ft_sort_string_tab	13
X	Esercizio 07 : ft_advanced_sort_string_tab	14

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 !!!

Capitolo II

Preambolo

Here's a little story :

(1982, California) Larry Walters of Los Angeles is one of the few to contend for the Darwin Awards and live to tell the tale. "I have fulfilled my 20-year dream," said Walters, a former truck driver for a company that makes TV commercials. "I'm staying on the ground. I've proved the thing works."

Larry's boyhood dream was to fly. But fates conspired to keep him from his dream. He joined the Air Force, but his poor eyesight disqualified him from the job of pilot. After he was discharged from the military, he sat in his backyard watching jets fly overhead.

He hatched his weather balloon scheme while sitting outside in his "extremely comfortable" Sears lawnchair. He purchased 45 weather balloons from an Army-Navy surplus store, tied them to his tethered lawnchair (dubbed the Inspiration I) and filled the four-foot diameter balloons with helium. Then, armed with some sandwiches, Miller Lite, and a pellet gun, he strapped himself into his lawnchair. He figured he would shoot to pop a few of the many balloons when it was time to descend.

Larry planned to sever the anchor and lazily float to a height of about 30 feet above the backyard, where he would enjoy a few hours of flight before coming back down. But things didn't work out quite as Larry planned.

When his friends cut the cord anchoring the lawnchair to his Jeep, he did not float lazily up to 30 feet. Instead he streaked into the LA sky as if shot from a cannon, pulled by the lift of 45 helium balloons, holding 33 cubic feet of helium each.

He didn't level off at 100 feet, nor did he level off at 1000 feet. After climbing and climbing, he leveled off at 16,000 feet.

At that height he felt he couldn't risk shooting any of the balloons, lest he unbalance the load and really find himself in trouble. So he stayed there, drifting cold and frightened with his beer and sandwiches, for more than 14 hours. He crossed the primary approach corridor of LAX, where startled Trans World Airlines and Delta Airlines pilots radioed in reports

of the strange sight.


Eventually he gathered the nerve to shoot a few balloons, and slowly descended. The hanging tethers tangled and caught in a power line, blacking out a Long Beach neighborhood for 20 minutes. Larry climbed to safety, where he was arrested by waiting members of the LAPD. As he was led away in handcuffs, a reporter dispatched to cover the daring rescue asked him why he had done it. Larry replied nonchalantly, "A man can't just sit around."

The Federal Aviation Administration was not amused. Safety Inspector Neal Savoy said, "We know he broke some part of the Federal Aviation Act, and as soon as we decide which part it is, a charge will be filed."

The moral of this story is Larry Walters should have stayed on his chair and learnt C....

Capitolo III

Esercizio 00 : ft_foreach

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

- Creare la funzione **ft_foreach** che esegue una funzione per ogni elemento di un array, seguendo l'ordine di quest'ultimo.
- Il prototipo è il seguente :

```
void      ft_foreach(int *tab, int length, void(*f)(int));
```

- La funzione **ft_foreach** potrebbe essere chiamata in questo modo per stampare a video l'array di int :

```
ft_foreach(tab, 1337, &ft_putnbr);
```

Capitolo IV

Esercizio 01 : ft_map


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

- Creare la funzione **ft_map** che esegue una funzione per ogni elemento di un array e restituisca un array contenente tutti valori restituiti.
- Il prototipo è il seguente :

```
int      *ft_map(int *tab, int length, int(*f)(int));
```


Capitolo V

Esercizio 02 : ft_any

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


- Creare la funzione **ft_any** che restituisce 1 se la funzione **f** restituisce qualcosa diverso da 0 per almeno un elemento dell'array passatole. In caso contrario deve restituire 0.
- La funzione verrà applicata seguendo l'ordine dell'array.
- Il prototipo è il seguente :

```
int ft_any(char **tab, int(*f)(char*));
```

- L'array sarà delimitato da un puntatore nullo.

Capitolo VI

Esercizio 03 : ft_count_if


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

- Creare la funzione `ft_count_if` che restituisce il numero di elementi dell'array che, passati alla funzione `f` non restituiscono 0.
- La funzione verrà applicata seguendo l'ordine dell'array.
- Il prototipo è il seguente :

```
int      ft_count_if(char **tab, int length, int(*f)(char*));
```

Capitolo VII

Esercizio 04 : ft_is_sort


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

- Create la funzione `ft_is_sort` che restituisce 1 se l'array è ordinato, 0 se non lo è.
- La funzione data come argomento restituisce o un intero negativo, se il primo argomento è minore del secondo, o 0, se uguali, o un intero positivo se il primo è maggiore del secondo.
- Il prototipo è il seguente :

```
int ft_is_sort(int *tab, int length, int(*f)(int, int));
```

Capitolo VIII

Esercizio 05 : do-op

	Esercizio 05
	do-op
	Cartella per la consegna : <i>ex05/</i>
	File da consegnare : Your program files
	Funzioni permesse : write

- Creare il program do-op.
- Il programma sarà eseguito con tre argomenti: `do-op valore1 operatore valore2`
- Ad esempio :

```
$> ./do-op 42 "+" 21
63
$>
```

- Dovreste usare un array di puntatori per l'operator.
- In caso di un operatore non valido il programma deve stampare 0.
- do-op non stampa niente se il numero di argomenti non è valido.
- Gli operatori validi sono: '+' '-' '/' '*' and '%'
- In caso di divisione per 0:

```
Stop : division by zero
```

- In caso di modulo per 0:


```
Stop : modulo by zero
```

- Ecco un esempio dei test effettuati da Moulinette :

```
$> make clean
$> make
$> ./do-op
$> ./do-op 1 + 1
2
$> ./do-op 42amis - ---20toto12
62
$> ./do-op 1 p 1
0
$> ./do-op 1 + toto3
1
$>
$> ./do-op toto3 + 4
4
$> ./do-op foo plus bar
0
$> ./do-op 25 / 0
Stop : division by zero
$> ./do-op 25 % 0
Stop : modulo by zero
$>
```

Capitolo IX

Esercizio 06 : ft_sort_string_tab


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

- Create the function `ft_sort_string_tab`, by `ascii` order the strings in `tab`.
- `tab` will be null terminated
- The sorting will be performed by exchanging the array's pointers.
- Il prototipo è il seguente :

```
void ft_sort_string_tab(char **tab);
```

Capitolo X

Esercizio 07 : ft_advanced_sort_string_tab

	Esercizio 07
	ft_advanced_sort_string_tab
	Cartella per la consegna : ex07/
	File da consegnare : ft_advanced_sort_string_tab.c
	Funzioni permesse : Nessuna

- Creare la funzione `ft_advanced_sort_string_tab` che ordina il primo parametro a seconda del valore restituito dalla funzione passata come secondo parametro
- L'ordinamento sarà effettuato scambiando i puntatori nell'array.
- `tab` sarà terminato da un puntatore nullo(null terminated)
- Il prototipo è il seguente :

```
void ft_advanced_sort_string_tab(char **tab, int(*cmp)(char *, char *));
```



Una chiamata di `ft_advanced_sort_string_tab()` con `ft_strcmp` come secondo parametro avrà come valore restituito lo stesso di `ft_sort_string_tab()`.