



# Piscina C

## C 05

*Sommario: Questo documento tratta il modulo C 05 della Piscina C @ 42.*

# Indice

I	Istruzioni	2
II	Preambolo	4
III	Esercizio 00 : ft_iterative_factorial	6
IV	Esercizio 01 : ft_recursive_factorial	7
V	Esercizio 02 : ft_iterative_power	8
VI	Esercizio 03 : ft_recursive_power	9
VII	Esercizio 04 : ft_fibonacci	10
VIII	Esercizio 05 : ft_sqrt	11
IX	Esercizio 06 : ft_is_prime	12
X	Esercizio 07 : ft_find_next_prime	13
XI	Exercise 08 : The Ten Queens	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 !!!



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

# Capitolo II

## Preambolo

Here are some lyrics extract from the Harry Potter saga:

Oh you may not think me pretty,  
But don't judge on what you see,  
I'll eat myself if you can find  
A smarter hat than me.

You can keep your bowlers black,  
Your top hats sleek and tall,  
For I'm the Hogwarts Sorting Hat  
And I can cap them all.

The Sorting Hat, stored in the Headmaster's Office.  
There's nothing hidden in your head  
The Sorting Hat can't see,  
So try me on and I will tell you  
Where you ought to be.

You might belong in Gryffindor,  
Where dwell the brave at heart,  
Their daring, nerve, and chivalry  
Set Gryffindors apart;

You might belong in Hufflepuff,  
Where they are just and loyal,  
Those patient Hufflepuffs are true  
And unafraid of toil;

Or yet in wise old Ravenclaw,  
If you've a ready mind,  
Where those of wit and learning,  
Will always find their kind;

Or perhaps in Slytherin  
You'll make your real friends,  
Those cunning folks use any means


To achieve their ends.

So put me on! Don't be afraid!  
And don't get in a flap!  
You're in safe hands (though I have none)  
For I'm a Thinking Cap!

Unfortunately, this subject's got nothing to do with the Harry Potter saga, which is too bad, because your exercises won't be done by magic.

# Capitolo III

## Esercizio 00 : ft\_iterative\_factorial


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

- Creare una funzione iterativa che restituisca il fattoriale del numero passato come parametro.
- Se il parametro non è valido la funzione restituirà 0.
- Nel caso di Overflow il risultato non è definito.
- Il prototipo è il seguente :

```
int ft_iterative_factorial(int nb);
```

# Capitolo IV

## Esercizio 01 : ft\_recursive\_factorial

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


- Creare una funzione ricorsiva che restituisca il fattoriale del numero passato come parametro.
- Se il parametro non è valido la funzione restituirà 0.
- Nel caso di Overflow il risultato non è definito.
- Il prototipo è il seguente :

```
int ft_recursive_factorial(int nb);
```



# Capitolo V

## Esercizio 02 : ft\_iterative\_power


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

- Creare una funzione iterativa che restituisca il valore di un numero **nb** elevato **power**. Se il parametro non è valido la funzione restituirà 0. Nel caso di Overflow il risultato non è definito.
- Nel caso 0 elevato 0 la funzione restituisce 1.
- Il prototipo è il seguente :

```
int ft_iterative_power(int nb, int power);
```

# Capitolo VI

## Esercizio 03 : ft\_recursive\_power


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

- Creare una funzione ricorsiva che restituisca il valore di un numero **nb** elevato **power**. Se il parametro non è valido la funzione restituirà 0. Nel caso di Overflow il risultato non è definito.
- Nel caso 0 elevato 0 la funzione restituisce 1.
- Il prototipo è il seguente :

```
int ft_recursive_power(int nb, int power);
```

# Capitolo VII

## Esercizio 04 : ft\_fibonacci

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


- Creare una funzione `ft_fibonacci` che ritorni l'elemento **n** della sequenza di Fibonacci. Per index 0 si intende il primo elemento. Considereremo una serie di Fibonacci che inizia così: 0, 1, 1, 2.
- Nel caso di Overflow il risultato non è definito
- Il prototipo è il seguente :

```
int ft_fibonacci(int index);
```

- `ft_fibonacci` sarà ovviamente ricorsiva.
- Per index minore di 0, la funzione restituirà -1.

# Capitolo VIII

## Esercizio 05 : ft\_sqrt


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

- Creare una funzione che restituisca la radice quadrata di un numero.
- Se la radice è un numero irrazionale, la funzione restituirà 0.
- Il prototipo è il seguente :

```
int ft_sqrt(int nb);
```

# Capitolo IX

## Esercizio 06 : ft\_is\_prime

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

- Creare una funzione che restituisca 1, se il numero passato come parametro è primo, o 0 se non lo è.
- Il prototipo è il seguente :


```
int ft_is_prime(int nb);
```



0 e 1 non sono numeri primi.

# Capitolo X

## Esercizio 07 : ft\_find\_next\_prime


	Esercizio 07
	ft_find_next_prime
	Cartella per la consegna : <i>ex07/</i>
	File da consegnare : <b>ft_find_next_prime.c</b>
	Funzioni permesse : Nessuna

- Creare una funzione che restituisca il primo numero primo maggiore o uguale al numero passatole come parametro.
- Il prototipo è il seguente :

```
int ft_find_next_prime(int nb);
```

# Capitolo XI

## Exercise 08 : The Ten Queens

	Esercizio 08
	The Ten Queens
	Cartella per la consegna : <i>ex08/</i>
	File da consegnare : <b>ft_ten_queens_puzzle.c</b>
	Funzioni permesse : <b>write</b>

- Creare una funzione che stampi a video tutte le possibili posizioni che possono assumere dieci regine su una scacchiera 10x10, senza che nessuna possa mangiarne un'altra in una sola mossa.
- Per risolvere il problema dovrete ricorrere ad una funzione ricorsiva.
- Il prototipo è il seguente :

```
int ft_ten_queens_puzzle(void);
```

- Questo sarà l'output :

```
$>./a.out | cat -e
0257948136$
0258693147$
...
4605713829$
4609582731$
...
9742051863$
$>
```

- Le sequenze indicano, da sinistra a destra, la posizione della regina sulla n colonna: la prima cifra indicherà la posizione della prima regina sulla prima colonna, la cifra n indicherà la posizione della regina n sulla colonna n.
- La funzione restituirà il numero totale di soluzioni trovate.