



# Piscine C

Jour 04

Staff 42 [piscine@42.fr](mailto:piscine@42.fr)

*Résumé: Ce document est le sujet du jour 04 de la piscine C de 42.*

# Table des matières

I	Consignes	2
II	Préambule	4
III	Exercice 00 : ft_iterative_factorial	5
IV	Exercice 01 : ft_recursive_factorial	6
V	Exercice 02 : ft_iterative_power	7
VI	Exercice 03 : ft_recursive_power	8
VII	Exercice 04 : ft_fibonacci	9
VIII	Exercice 05 : ft_sqrt	10
IX	Exercice 06 : ft_is_prime	11
X	Exercice 07 : ft_find_next_prime	12
XI	Exercice 08 : Les huit dames	13
XII	Exercice 09 : Les huit dames 2	14

# Chapitre I

## Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices.
- Vos exercices seront corrigés par vos camarades de piscine.
- En plus de vos camarades, vous serez corrigés par un programme appelé la Moulinette.
- La Moulinette est très stricte dans sa notation. Elle est totalement automatisée. Il est impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les surprises.
- La Moulinette n'est pas très ouverte d'esprit. Elle ne cherche pas à comprendre le code qui ne respecte pas la Norme.
- L'utilisation d'une fonction interdite est un cas de triche. Toute triche est sanctionnée par la note de -42.
- Si `ft_putchar()` est une fonction autorisée, nous compilerons avec notre `ft_putchar.c`.
- Vous ne devrez rendre une fonction `main()` que si nous vous demandons un programme.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- La Moulinette compile avec les flags `-Wall -Wextra -Werror`.
- Si votre programme ne compile pas, vous aurez 0.
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.

- Votre manuel de référence s'appelle **Google / man / Internet / ....**
- Pensez à discuter sur le forum Piscine de votre Intra !
- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne sont pas autrement précisées dans le sujet...
- Réfléchissez. Par pitié, par Odin ! Nom d'une pipe.

# Chapitre II


## Préambule

Voici les 2399 premières décimales de pi :

3,141592653589793238462643383279502884197169399375105820974944592307816406286208  
99862803482534211706798214808651328230664709384460955058223172535940812848111745  
02841027019385211055596446229489549303819644288109756659334461284756482337867831  
65271201909145648566923460348610454326648213393607260249141273724587006606315588  
17488152092096282925409171536436789259036001133053054882046652138414695194151160  
94330572703657595919530921861173819326117931051185480744623799627495673518857527  
24891227938183011949129833673362440656643086021394946395224737190702179860943702  
77053921717629317675238467481846766940513200056812714526356082778577134275778960  
91736371787214684409012249534301465495853710507922796892589235420199561121290219  
60864034418159813629774771309960518707211349999998372978049951059731732816096318  
59502445945534690830264252230825334468503526193118817101000313783875288658753320  
83814206171776691473035982534904287554687311595628638823537875937519577818577805  
32171226806613001927876611195909216420198938095257201065485863278865936153381827  
96823030195203530185296899577362259941389124972177528347913151557485724245415069  
59508295331168617278558890750983817546374649393192550604009277016711390098488240  
12858361603563707660104710181942955596198946767837449448255379774726847104047534  
64620804668425906949129331367702898915210475216205696602405803815019351125338243  
00355876402474964732639141992726042699227967823547816360093417216412199245863150  
30286182974555706749838505494588586926995690927210797509302955321165344987202755  
96023648066549911988183479775356636980742654252786255181841757467289097777279380  
00816470600161452491921732172147723501414419735685481613611573525521334757418494  
68438523323907394143334547762416862518983569485562099219222184272550254256887671  
79049460165346680498862723279178608578438382796797668145410095388378636095068006  
42251252051173929848960841284886269456042419652850222106611863067442786220391949  
45047123713786960956364371917287467764657573962413890865832645995813390478027590  
09946576407895126946839835259570982582262052248940772671947826848260147699090264  
01363944374553050682034962524517493996514314298091906592509372216964615157098583  
87410597885959772975498930161753928468138268683868942774155991855925245953959431  
04997252468084598727364469584865383673622262609912460805124388439045124413654976  
27807977156914359977001296160894416948685558484063534220722258284886481584560285

# Chapitre III

## Exercice 00 : ft\_iterative\_factorial

	Exercice : 00
ft_iterative_factorial	
Dossier de rendu : ex00/	
Fichiers à rendre : ft_iterative_factorial.c	
Fonctions Autorisées : Aucune	
Remarques : n/a	


- Écrire une fonction itérative qui renvoie un nombre. Ce nombre est le résultat de l'opération factorielle à partir du nombre passé en paramètre.
- En cas d'erreur, la fonction devra retourner 0.
- Elle devra être prototypée de la façon suivante :

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

- Votre fonction doit donner son résultat en moins de deux secondes.

# Chapitre IV

## Exercice 01 : ft\_recursive\_factorial


	Exercice : 01
ft_recursive_factorial	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <b>ft_recursive_factorial.c</b>	
Fonctions Autorisées : <b>Aucune</b>	
Remarques : <b>n/a</b>	

- Écrire une fonction récursive qui renvoie la factorielle du nombre passé en paramètre.
- Elle doit gérer les mêmes cas que la fonction précédente.
- Elle devra être prototypée de la façon suivante :

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

# Chapitre V

## Exercice 02 : ft\_iterative\_power

	Exercice : 02
ft_iterative_power	
Dossier de rendu : ex02/	
Fichiers à rendre : ft_iterative_power.c	
Fonctions Autorisées : Aucune	
Remarques : n/a	

- Écrire une fonction itérative qui renvoie une puissance d'un nombre. Une puissance négative renverra 0.
- Elle devra être prototypée de la façon suivante :


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

- Votre fonction doit donner son résultat en moins de deux secondes.



# Chapitre VI

## Exercice 03 : ft\_recursive\_power


	Exercice : 03
ft_recursive_power	
Dossier de rendu : ex03/	
Fichiers à rendre : ft_recursive_power.c	
Fonctions Autorisées : Aucune	
Remarques : n/a	

- Écrire une fonction récursive qui renvoie une puissance d'un nombre.
- Elle doit gérer les mêmes cas que la fonction précédente.
- Elle devra être prototypée de la façon suivante :

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

# Chapitre VII

## Exercice 04 : ft\_fibonacci

	Exercice : 04
	ft_fibonacci
	Dossier de rendu : <i>ex04/</i>
	Fichiers à rendre : <b>ft_fibonacci.c</b>
	Fonctions Autorisées : <b>Aucune</b>
	Remarques : <b>n/a</b>


- Écrire une fonction `ft_fibonacci` qui renvoie le `n`-ième élément de la suite de Fibonacci, le premier élément étant à l'index 0. Nous considererons que la suite de Fibonacci commence par 0, 1, 1, 2.
- Elle devra être prototypée de la façon suivante :

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

- Evidement, `ft_fibonacci` devra être recursive.
- Si `index` est négatif, la fonction renverra -1.

# Chapitre VIII

## Exercice 05 : ft\_sqrt

	Exercice : 05
ft_sqrt	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : <b>ft_sqrt.c</b>	
Fonctions Autorisées : Aucune	
Remarques : n/a	


- Écrire une fonction qui renvoie la racine carrée entière d'un nombre si elle existe, 0 si la racine carrée n'est pas entière.
- Elle devra être prototypée de la façon suivante :

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

- Votre fonction doit donner son résultat en moins de deux secondes.

# Chapitre IX

## Exercice 06 : ft\_is\_prime

	Exercice : 06
	ft_is_prime
Dossier de rendu : <i>ex06/</i>	
Fichiers à rendre : <code>ft_is_prime.c</code>	
Fonctions Autorisées : Aucune	
Remarques : n/a	

- Écrire une fonction qui renvoie 1 si le nombre est premier et 0 si le nombre ne l'est pas.
- Elle devra être prototypée de la façon suivante :

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


- Votre fonction doit donner son résultat en moins de deux secondes.



0 et 1 ne sont pas des nombres premiers.

# Chapitre X

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

	Exercice : 07
ft_find_next_prime	
Dossier de rendu : ex07/	
Fichiers à rendre : ft_find_next_prime.c	
Fonctions Autorisées : Aucune	
Remarques : n/a	


- Écrire une fonction qui renvoie le nombre premier immédiatement supérieur ou égal au nombre passé en paramètre.
- Elle devra être prototypée de la façon suivante :

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

- Votre fonction doit donner son résultat en moins de deux secondes.

# Chapitre XI

## Exercice 08 : Les huit dames

	Exercice : 08
Les huit dames 1	
Dossier de rendu : <i>ex08/</i>	
Fichiers à rendre : <code>ft_eight_queens_puzzle.c</code>	
Fonctions Autorisées : Aucune	
Remarques : n/a	


- Le but de ce jeu est de placer huit dames sur un échiquier sans qu'elles ne puissent s'atteindre en un seul coup.
- Rafraîchissez-vous la mémoire sur les règles des échecs.
- Bien entendu, on utilisera la récursivité pour résoudre ce problème.
- Écrire une fonction qui renvoie le nombre de possibilités de placer les huit dames sur l'échiquier sans qu'elles ne puissent s'atteindre.
- Elle devra être prototypée de la façon suivante :

```
int ft_eight_queens_puzzle(void);
```

- Votre fonction doit donner son résultat en moins de deux secondes.

# Chapitre XII

## Exercice 09 : Les huit dames 2

	Exercice : 09
Les huit dames 2	
Dossier de rendu : <i>ex09/</i>	
Fichiers à rendre : <code>ft_eight_queens_puzzle_2.c</code>	
Fonctions Autorisées : <code>ft_putchar</code>	
Remarques : n/a	

- Écrire une fonction qui affiche toutes les possibilités de placer les huit dames sur l'échiquier sans qu'elles ne puissent s'atteindre.
- La recursivité devra être utilisée.
- Elle devra être prototypée de la façon suivante :

```
void ft_eight_queens_puzzle_2(void);
```

- L'affichage se fera de la façon suivante :

```
$> ./a.out  
15863724  
16837425  
17468253  
...
```

- La suite se lit de gauche à droite. Le premier chiffre correspond à la position de la première dame dans la première colonne (l'index commençant à 1). Le *énième* chiffre correspond à la position de la *énième* dame dans la *énième* colonne.
- Il y a un saut de ligne après la dernière solution.
- Votre fonction doit donner son résultat en moins de deux secondes.