

# Piscine C Jour 04

Staff 42 piscine@42.fr

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

# Table des matières

Ι	Consignes	2
II	Préambule	4
III	Exercice 00 : ft_iterative_factorial	5
IV	Exercice 01 : ft_recursive_factorial	6
$\mathbf{V}$	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
$\mathbf{X}$	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 <u>ne devez</u> laisser dans votre répertoire <u>aucun</u> autre fichier que ceux explicitement specifiés 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.

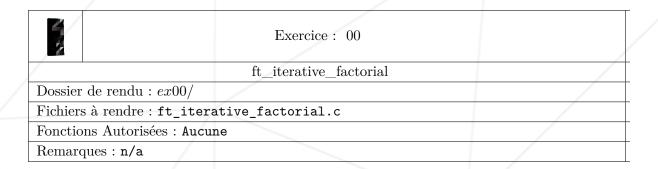
# Chapitre II Préambule

Voici les 2399 premières décimales de pi :

3,141592653589793238462643383279502884197169399375105820974944592307816406286208

#### Chapitre III

#### Exercice 00: ft\_iterative\_factorial

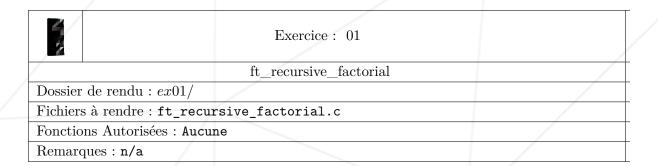


- Écrire une fonction itérative qui renvoie un nombre. Ce nombre est le résultat de l'opération factorielle à partir du nombre passé en paramètre.
- En cas d'erreur, la fonction devra retourner 0.
- Elle devra être prototypée de la façon suivante :

int ft\_iterative\_factorial(int nb);

#### Chapitre IV

# Exercice 01: ft\_recursive\_factorial



- 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

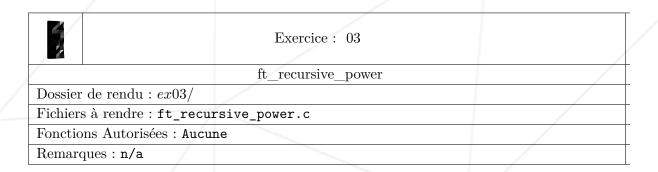
	Exercice: 02	
	ft_iterative_power	
Dossier de rendu : $ex02/$		
Fichiers à rendre : ft_iterative_power.c		
Fonctions Autorisées : Aucune		
Remarques : n/a		

- Écrire une fonction itérative qui renvoie une puissance d'un nombre. Une puissance < 0 renverra 0. Les overflows ne devront pas être gerés.
- Elle devra être prototypée de la façon suivante :

int ft\_iterative\_power(int nb, int power);

#### Chapitre VI

# Exercice 03: ft\_recursive\_power



- Écrire une fonction récursive qui renvoie une puissance d'un nombre.
- $\bullet\,$  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 : $ex04/$		
Fichiers à rendre : ft_fibonacci.c		
Fonctions Autorisées : Aucune		
Remarques : n/a		

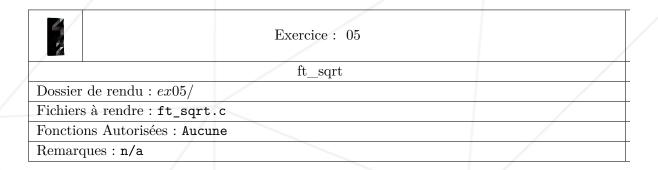
- Écrire une fonction ft\_fibonacci qui renvoie le n-ième élément de la suite de Fibonacci, le premier élément étant à l'index 0. Nous considererons que la suite de Fibonacci commence par 0, 1, 1, 2.
- Elle devra être prototypée de la façon suivante :

int ft\_fibonacci(int index);

- $\bullet$  Evidement, ft\_fibonacci devra être recursive.
- Si index est négatif, la fonction renverra -1.

# Chapitre VIII

Exercice 05: ft\_sqrt

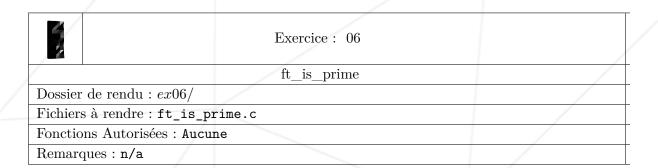


- Écrire une fonction qui renvoie la racine carrée entière d'un nombre si elle existe, 0 si la racine carrée n'est pas entière.
- Elle devra être prototypée de la façon suivante :

int ft\_sqrt(int nb);

#### Chapitre IX

Exercice 06: ft\_is\_prime



- $\bullet$  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);

#### Chapitre XI

# Exercice 08: Les huit dames

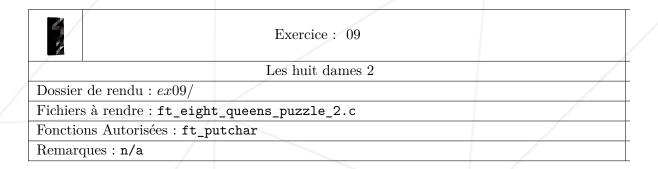
	Exercice: 08	
/	Les huit dames 1	
Dossier de rendu : $ex08/$		
Fichiers à rendre : ft_eight_queens_puzzle.c		
Fonctions Autorisées : Aucune		
Remarques : n/a		

- 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 recursivité 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);

#### Chapitre XII

#### Exercice 09: Les huit dames 2



- 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 commenant 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.