



# 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

<b>I</b>	<b>Consignes</b>	<b>2</b>
<b>II</b>	<b>Préambule</b>	<b>4</b>
<b>III</b>	<b>Exercice 00 : ft_iterative_factorial</b>	<b>6</b>
<b>IV</b>	<b>Exercice 01 : ft_recursive_factorial</b>	<b>7</b>
<b>V</b>	<b>Exercice 02 : ft_iterative_power</b>	<b>8</b>
<b>VI</b>	<b>Exercice 03 : ft_recursive_power</b>	<b>9</b>
<b>VII</b>	<b>Exercice 04 : ft_fibonacci</b>	<b>10</b>
<b>VIII</b>	<b>Exercice 05 : ft_sqrt</b>	<b>11</b>
<b>IX</b>	<b>Exercice 06 : ft_is_prime</b>	<b>12</b>
<b>X</b>	<b>Exercice 07 : ft_find_next_prime</b>	<b>13</b>
<b>XI</b>	<b>Exercice 08 : Les huit dames</b>	<b>14</b>
<b>XII</b>	<b>Exercice 09 : Les huit dames 2</b>	<b>15</b>

# 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. La Moulinette utilise le programme **norminette** pour vérifier la norme de vos fichiers. Comprendre par là qu'il est stupide de rendre un code qui ne passe pas la **norminette**.
- 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`, et utilise `gcc`.
- 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.



Pour cette journée, la norminette doit être lancée avec le flag `-R CheckForbiddenSourceHeader`. La moulinette l'utilisera aussi.

# Chapitre II

## Préambule

Voici des paroles extraite du premier livre de la saga Harry Potter :

Je n'suis pas d'une beauté suprême  
Mais faut pas s'fier à ce qu'on voit  
Je veux bien me manger moi-même  
Si vous trouvez plus malin qu'moi.

Les hauts-d'forme, les chapeaux splendides,  
Font pâl'figure auprès de moi  
Car à Poudlard, quand je décide,  
Chacun se soumet à mon choix.

Rien ne m'échapp' rien ne m'arrête  
Le Choixpeau a toujours raison  
Mettez-moi donc sur votre tête  
Pour connaître votre maison.

Si vous allez à Gryffondor  
Vous rejoindrez les courageux,  
Les plus hardis et les plus forts  
Sont rassemblés en ce haut lieu.

Si à Poufsouffle vous allez,  
Comme eux vous s'rez juste et loyal  
Ceux de Poufsouffle aiment travailler  
Et leur patience est proverbiale.

Si vous êtes sage et réfléchi  
Serdaigle vous accueillera peut-être  
Là-bas, ce sont des érudits  
Qui ont envie de tout connaître.


Vous finirez à Serpentard  
Si vous êtes plutôt malin,  
Car ceux-là sont de vrais roublards  
Qui parviennent toujours à leurs fins.

Sur ta tête pose-moi un instant  
Et n'aie pas peur, reste serein  
Tu seras en de bonnes mains  
Car je suis un chapeau pensant !

Ce sujet n'a, malheureusement, rien à voir avec la série **Harry Potter**, et c'est dommage, parce que votre rendu ne sera pas fait par **magie**.

# 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	


- 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 : Aucune	


- 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 : <i>ex02/</i>	
Fichiers à rendre : <b>ft_iterative_power.c</b>	
Fonctions Autorisées : Aucune	


- Écrire une fonction itérative qui renvoie une puissance d'un nombre. Une puissance inférieur à 0 renverra 0. Les overflows ne devront pas être gérés.
- 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 : <i>ex03/</i>	
Fichiers à rendre : <b>ft_recursive_power.c</b>	
Fonctions Autorisées : Aucune	

- 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 : Aucune


- 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);
```

- Évidemment, `ft_fibonacci` devra être récursive.
- Si `index` est inférieur à 0, 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	


- 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	

- 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 : <i>ex07/</i>	
Fichiers à rendre : <code>ft_find_next_prime.c</code>	
Fonctions Autorisées : Aucune	


- 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	


- 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>	

- Écrire une fonction qui affiche toutes les possibilités de placer les huit dames sur l'échiquier sans qu'elles ne puissent s'atteindre.
- La récursivité 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.