

Piscine C C 05

Résumé: ce document est le sujet du module C 05 de la piscine C de 42.

Table des matières

•	Completion	_
II	Préambule	4
III	Exercice 00 : ft_iterative_factorial	6
IV	Exercice 01 : ft_recursive_factorial	7
\mathbf{V}	Exercice 02 : ft_iterative_power	8
VI	Exercice 03 : ft_recursive_power	9
VII	Exercice 04 : ft_fibonacci	10
VIII	Exercice 05 : ft_sqrt	11
IX	Exercice 06 : ft_is_prime	12
\mathbf{X}	Exercice 07 : ft_find_next_prime	13
XI	Exercice 08 : Les dix dames	14

Chapitre I

Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Relisez bien le sujet avant de rendre vos exercices. A tout moment le sujet peut changer.
- 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.
- 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.
- L'utilisation d'une fonction interdite est un cas de triche. Toute triche est sanctionnée par la note de -42.
- Vous ne devrez rendre une fonction main() que si nous vous demandons un <u>programme</u>.
- La Moulinette compile avec les flags -Wall -Wextra -Werror, et utilise gcc.
- 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, ainsi que sur le slack de votre Piscine!
- 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 connaitre 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.

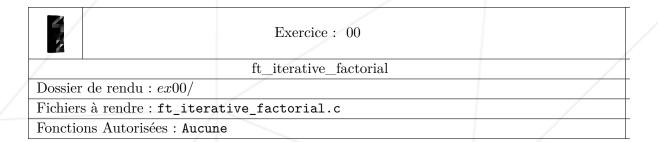
Si vous êtes sage et réfléchi Serdaigle vous accueillera peut-être Là-bas, ce sont des érudits Qui ont envie de tout connaître.

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

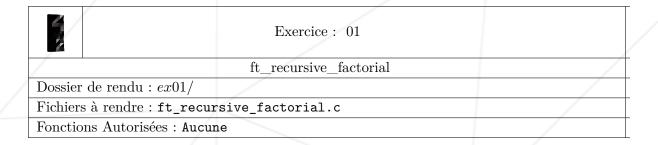


- Écrire une fonction itérative qui renvoie un nombre. Ce nombre est le résultat de l'opération factorielle à partir du nombre passé en paramètre.
- Si l'argument n'est pas valide, la fonction doit renvoyer 0.
- Il ne faut pas gerer les "int overflow", le retour de la fonction sera indefini.
- $\bullet\,$ Elle devra être prototypée de la façon suivante :

int ft_iterative_factorial(int nb);

Chapitre IV

Exercice 01: ft_recursive_factorial

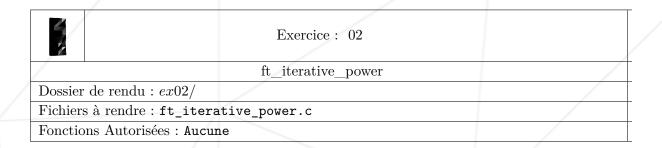


- Écrire une fonction récursive qui renvoie la factorielle du nombre passé en paramètre.
- Si l'argument n'est pas valide, la fonction doit renvoyer 0.
- Il ne faut pas gerer les "int overflow", le retour de la fonction sera indefini.
- $\bullet\,$ Elle devra être prototypée de la façon suivante :

int ft_recursive_factorial(int nb);

Chapitre V

Exercice 02: ft_iterative_power

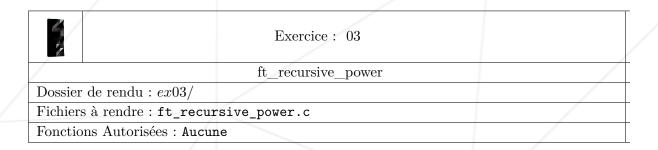


- Écrire une fonction itérative qui renvoie une puissance d'un nombre. Une puissance inferieur à 0 renverra 0.
- Comme il n'y a pas de concensus sur 0 puissance 0, nous considererons que le resultat sera 1.
- Il ne faut pas gerer les "int overflow", le retour de la fonction sera indefini.
- Elle devra 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.
- Comme il n'y a pas de concensus sur 0 puissance 0, nous considererons que le resultat sera 1.
- Il ne faut pas gerer les "int overflow", le retour de la fonction sera indefini.
- 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_fibo		
Fonctions Autorisées : Auci	/	

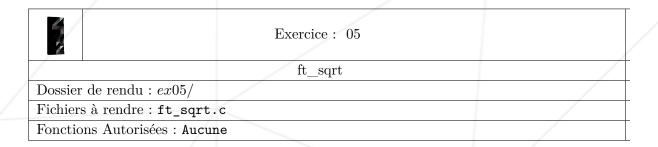
- Écrire une fonction ft_fibonacci qui renvoie le n-ième élément de la suite de Fibonacci, le premier élément étant à l'index 0. Nous considererons que la suite de Fibonacci commence par 0, 1, 1, 2.
- Les overflows ne devront pas être gerés.
- Elle devra être prototypée de la façon suivante :

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

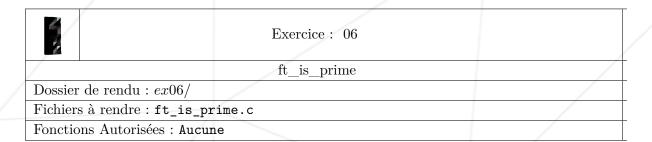


- Écrire une fonction qui renvoie la racine carrée entière d'un nombre si elle existe, 0 si la racine carrée n'est pas entière.
- Elle devra être prototypée de la façon suivante :

int ft_sqrt(int nb);

Chapitre IX

Exercice 06: ft_is_prime



- 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 :

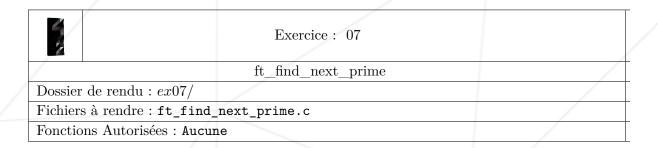
int ft_is_prime(int nb);



0 et 1 ne sont pas des nombres premiers.

Chapitre X

Exercice 07: ft_find_next_prime

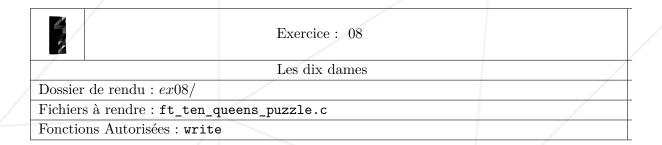


- Écrire une fonction qui renvoie le nombre premier immédiatement supérieur ou égal au nombre passé en paramètre.
- Elle devra être prototypée de la façon suivante :

int ft_find_next_prime(int nb);

Chapitre XI

Exercice 08: Les dix dames



- Écrire une fonction qui affiche toutes les possibilités de placer dix dames sur un échiquier de 10x10 sans qu'elles ne puissent s'atteindre en un seul coup.
- La recursivité devra être utilisée.
- La valeur de retour de votre fonction devra être le nombre de solutions affichées
- Elle devra être prototypée de la façon suivante :

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

• L'affichage se fera de la façon suivante :

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

• 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 0). Le énième chiffre correspond à la position de la énième dame dans la énième colonne.