



Piscine - C - Tek1

Sujet Jour 05

Responsables Astek astek_resp@epitech.eu



Table des matières

Consignes	2
0 - winner	3
1 - my_factorielle_it	4
2 - my_factorielle_rec	5
3 - my_power_it	6
4 - my_power_rec	7
5 - Les 8 reines 1	8
6 - Les 8 reines 2	9
7 - my_square_root	10
8 - my_is_prime	11
09 - my_find_prime_sup	12



Consignes

- Le sujet peut changer jusqu'à une heure avant le rendu.
- Vos exercices doivent être à la norme.
- Vous ne devez avoir de `main()` dans aucun fichier de votre repertoire de rendu.
- Pour chaque repertoire de chaque exercice nous allons compiler vos fichiers avec la commande `cc -c *.c`, ce qui va générer tous les fichiers `.o` que nous allons ensuite linker un par un en y ajoutant notre `main.c` et notre `my_putchar.c` :

```
$> cc -c *.c
$> cc *.o ~moulinette/main\_ex\_01.o ~moulinette/my\_putchar.o -o ex01
$> ./ex01
[...]
```

- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
Si un seul de vos fichiers empêche la compilation avec `*.c`, la moulinette ne pourra pas vous corriger et vous aurez 0. Vous avez donc tout intérêt à effacer vos rendus d'exercices ne fonctionnant pas.
- Vous n'avez le droit qu'à la fonction `my_putchar` pour faire les exercices qui suivent. Cette fonction sera fournie, donc :
 - vous ne devez pas avoir lors du rendu de fichier `my_putchar.c`
 - la fonction `my_putchar` ne doit être mise dans aucun des fichiers rendus
- Pensez à en discuter sur le forum piscine !
- Travaillez en local !
C'est-à-dire que pour chaque exercice vous devez le compiler sur votre compte linux puis, une fois qu'il fonctionne, le copier sur votre compte AFS.
Ceci dans le simple but de ne pas surcharger les serveurs car vous êtes nombreux.



Indices Faites-vous un script shell pour copier vos fichiers sur l'AFS

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_05`



0 - winner

- Le gagnant des mail d'hier est desoul_b <benoit.desoulle@epitech.eu>
- n'hésitez pas a le feliciter :)
- son mail est visible ici pour les curieux :
[http ://perso.epitech.eu/~plassi_y/mail.txt](http://perso.epitech.eu/~plassi_y/mail.txt)



1 - my_factorielle_it

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

```
1 int my_factorielle_it(int nb);
```

- Vous devez gérer si la factorielle du nombre fait un overflow (qui est une erreur).
- Rendu :

/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/my_factorielle_it.c



Votre fonction doit donner son résultat en moins de 2 secondes



Indices

- `factorielle(0) = 1`
- Si `(n < 0)` alors `factorielle(n) = 0`
- `n puissance 0 = 1`
- Si `(p < 0)` alors `n puissance p = 0`



2 - my__factorielle__rec

- Écrire une fonction récursive qui renvoie la factorielle du nombre passé en paramètre.
- Elle devra être prototypée de la façon suivante :

```
1  int my_factorielle_rec(int nb);
```

- Rendu :

`/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/my_factorielle_rec.c`



Elle doit gérer les mêmes choses que la fonction `my_factorielle_it`



3 - my__power__it

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

```
1 int my_power_it(int nb, int power);
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/my_power_it.c`



Votre fonction doit donner son résultat en moins de 2 secondes



4 - my__power__rec

- Écrire une fonction récursive qui renvoie la puissance d'un nombre.
- Elle devra être prototypée de la façon suivante :

```
1 int my_power_rec(int nb, int power);
```

- Rendu :
/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/my_power_rec.c



Elle doit gérer les mêmes choses que la fonction my_power_it



5 - Les 8 reines 1

- Le but de ce jeu est de placer 8 reines sur un échiquier sans qu'elles ne puissent s'atteindre en un seul coup.
- Détails techniques :
 - Un échiquier, c'est un damier de 8x8 cases.
 - Une reine peut jouer en ligne, en colonne et en diagonale.
- Bien entendu, on utilisera la récursivité pour résoudre ce problème.
- Écrire une fonction qui renvoie le nombre de possibilités de placer les 8 reines sur l'échiquier sans qu'elles ne puissent s'atteindre.
- Elle devra être prototypée de la façon suivante :

```
1  int my_8r1();
```

- Rendu :
/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/ex_05/my_8r1.c



Votre fonction doit donner son résultat en moins de 2 secondes



6 - Les 8 reines 2

- Écrire une fonction qui affiche toutes les possibilités de placer les 8 reines sur l'échiquier sans qu'elles ne puissent s'atteindre.
- La récursivité devra être utilisée.
- Elle devra être prototypée de la façon suivante :

```
1  int my_8r2();
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/my_8r2.c`
- L'affichage se fera de la façon suivante (les résultats ci-dessous sont faux, ils ne servent qu'à illustrer l'affichage) :

```
1  15346872
2  13564287
3  ...
```

- Il y a un saut de ligne après la dernière solution du problème des 8 reines.



Votre fonction doit donner son résultat en moins de 2 secondes



7 - my_square_root

- 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 :

```
1  int my_square_root(int nb);
```

- Rendu :

`/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/my_square_root.c`



Votre fonction doit donner son résultat en moins de 2 secondes



8 - my_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 :

```
1  int my_is_prime(int nb);
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/my_is_prime.c`



Indices 0 et 1 ne sont pas premiers



Votre fonction doit donner son résultat en moins de 2 secondes



09 - my__find__prime__sup

- 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 :

```
1  int my_find_prime_sup(int nb)
```

- Rendu :

`/afs/epitech.net/users/group/login/rendu/piscine/Jour_05/my_find_prime_sup.c`



Votre fonction doit donner son résultat en moins de 2 secondes