

Piscine C C 07

Staff 42 piscine@42.fr

Résumé: Ce document est le sujet du module C 07 de la piscine C de 42.

Table des matières

1	Consignes	2
II	Préambule	4
III	Exercice 00 : ft_strdup	5
IV	Exercice 01 : ft_range	6
\mathbf{V}	Exercice 02 : ft_ultimate_range	7
VI	Exercice 03 : ft_strjoin	8
VII	Exercice 04 : ft_convert_base	9
VIII	Exercice 05 : ft_split	10

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.

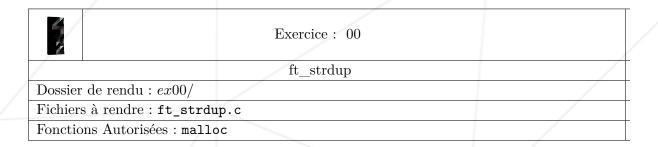
Chapitre II Préambule

Voici une liste des monstres que l'on peut trouver dans le célèbre Donjon de Naheulbeuk :

```
- Toutes sortes de morts-vivants ;
- Des araignées géantes ;
- Des orques ;
- Des gobelins ;
- Des trolls dans les souterrains ;
- Des sorciers ;
- Des guerriers maudits ;
- Des rats mutants ;
- Une bouteille d'huile ;
- Du papier toilette ;
- Deux éponges ;
- Des raviolis.
```

Chapitre III

Exercice 00: ft_strdup

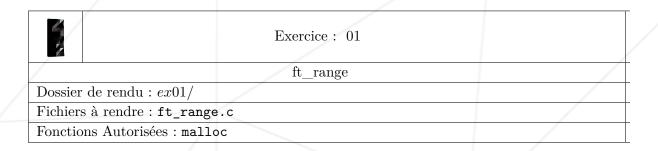


- Reproduire à l'identique le fonctionnement de la fonction strdup (man strdup).
- Elle devra être prototypée de la façon suivante :

char *ft_strdup(char *src);

Chapitre IV

Exercice 01: ft_range



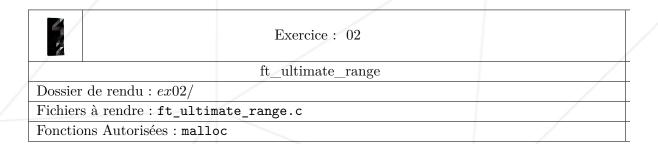
- Écrire une fonction ft_range qui retourne un tableau d'int. Ce tableau d'int contiendra toutes les valeurs entre min et max.
- Min inclu max exclu.
- Elle devra être prototypée de la façon suivante :

int *ft_range(int min, int max);

• Si la valeur min est supérieure ou égale à la valeur max, un pointeur nul sera retourné.

Chapitre V

Exercice 02: ft_ultimate_range



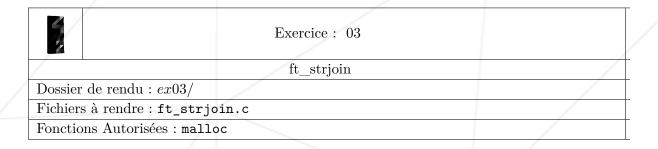
- Écrire une fonction ft_ultimate_range qui alloue et assigne un tableau d'int. Ce tableau d'int contiendra toutes les valeurs entre min et max.
- Min inclu max exclu.
- Elle devra être prototypée de la façon suivante :

int ft_ultimate_range(int **range, int min, int max);

- La taille de range sera retournée (ou -1 en cas de problème).
- Si la valeur min est supérieure ou égale à la valeur max, range pointera sur NULL et on renverra 0.

Chapitre VI

Exercice 03: ft_strjoin

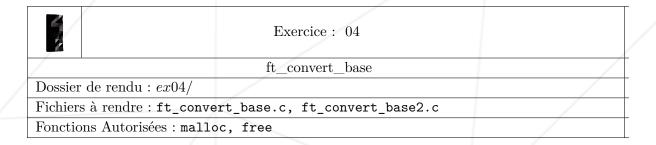


- Écrire une fonction qui va concatener l'ensemble des chaîne de caractères pointées par strs en les séparants à l'aide de sep.
- size représente la taille de strs.
- Si size vaut 0, il faut retourner une chaîne de caractères vide que l'on peut free().
- Elle devra être prototypée de la façon suivante :

char *ft_strjoin(int size, char **strs, char *sep);

Chapitre VII

Exercice 04: ft_convert_base

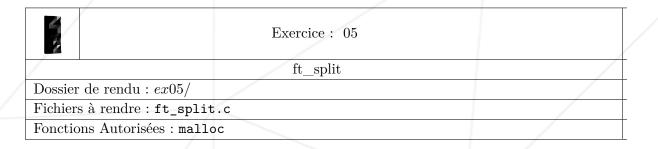


- Écrire une fonction qui renvoie le résultat de la conversion de la chaîne nbr exprimée en une base base_from dans une base base_to.
- nbr, base_from, base_to ne seront pas forcement modifiable.
- nbr suivra les mme règles que ft_atoi_base. Attention donc au '+', '-' et aux whitespaces.
- Le nombre representé par nbr tient dans un int.
- Si une base est incorrecte, la fonction renverra NULL.
- Le nombre retourné doit être préfixé seulement par un seul et unique '-' si c'est nécessaire, pas de whitespaces ou de '+'.
- Elle devra être prototypée de la façon suivante :

char *ft_convert_base(char *nbr, char *base_from, char *base_to);

Chapitre VIII

Exercice 05: ft_split



- Écrire une fonction qui découpe une chaîne de caractères en fonction d'une autre chaîne de caractères.
- Il faudra utiliser chaque caractère de la chaine charset comme séparateur.
- La fonction renvoie un tableau où chaque élément de celui ci contient l'adresse d'une chaîne de caractères comprise entre deux séparateur. Le dernier élement du tableau devra être égal à 0 pour marquer la fin du tableau.
- Il ne doit pas y avoir de chaîne vide dans votre tableau. Tirez-en les conclusions qui s'imposent.
- La chaîne qui sera transmise ne sera pas modifiable.
- Elle devra être prototypée de la façon suivante :

char **ft_split(char *str, char *charset);