



Piscine C

C 09

Résumé: Ce document est le sujet du module C 09 de la piscine C de 42.

Version:

Table des matières

I	Consignes	2
II	Préambule	4
III	Exercice 00 : libft	5
IV	Exercice 01 : Makefile	6
V	Exercice 02 : ft_split	8
VI	Rendu et peer-evaluation	9

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 programme.
- 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, 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

Dialogue issue du film The Big Lebowski :

Walter : Excuse-moi Smokey, t'as mordu la ligne : y'a faute.

Smokey : Mon cul ! Tu me mets huit, Duc...

Walter : Je te demande pardon, mets lui zéro, jeu suivant.

Smokey : Tu fais chier Walter, tu me mets huit Duc...

Walter : Smokey on est pas au Viet-nâm, on est au bowling, on joue selon les règles...

Le Duc : Allez, déconne pas Walter, on est là merde : son pied a légèrement mordu, il a un peu glissé... C'est qu'un sport, là !

Walter : Oui. Et il est homologué, c'est une partie qui compte pour le tournoi, j'ai pas raison ?

Smokey : Ouais, mais j'ai pas...

Walter : J'ai pas raison ?

Smokey : Ouais, mais j'ai pas mordu... Allez ! Vas-y Duc, mets moi un huit.

Walter : [Sort son flingue] Smokey mon ami, si t'as jamais souffert, tu vas comprendre...

Le Duc : Walter, fais pas le con !

Walter : [Menaçant] Vas-y, mets-toi un huit et tu vas comprendre !

Smokey : J'ai pas m...

Walter : Tu vas comprendre ta douleur, Smokey !

Smokey : Duc ? C'est ton partenaire...

Walter : Le monde est en train de DEVENIR CINGLÉ ! [Se lève] Y'A PERSONNE ICI À PART MOI QUI SE SOUCIE ENCORE DE RESPECTER LES RÈGLES ? METS-TOI ZÉRO !

Le Duc : Ils sont en train d'appeler les flics Walter remets ça dans...

Walter : [Braque son flingue sur Smokey] METS-TOI ZÉRO !

Le Duc : Range ça Walter...

Smokey : ...

Le Duc : Walter...


Walter : [Arme son flingue] TU CROIS PEUT-ÊTRE QUE JE PLAISANTE ? METS-TOI ZÉRO!

Smokey : Voilà, je me suis mis zéro... T'es content ? Espèce de malade !

Walter : [Se calme] ... C'est un sport homologué.

Chapitre III

Exercice 00 : libft

	Exercice : 00
	libft
	Dossier de rendu : <i>ex00/</i>
	Fichiers à rendre : <code>libft_creator.sh</code> , <code>ft_putchar.c</code> , <code>ft_swap.c</code> , <code>ft_putstr.c</code> , <code>ft_strlen.c</code> , <code>ft_strcmp.c</code>
	Fonctions Autorisées : <code>write</code>

- Créer votre bibliothèque `ft`. Elle s'appellera `libft.a`.
- Un shell script appelé `libft_creator.sh` compilera comme il le faut les fichiers sources et créera votre bibliothèque.
- Cette bibliothèque doit contenir toutes les fonctions suivantes :


```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

- Nous lancerons la commande suivante :

```
sh libft_creator.sh
```

Chapitre IV

Exercice 01 : Makefile

	Exercice : 01
Makefile	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : Makefile	
Fonctions Autorisées : Aucune	

- Écrire le **Makefile** qui compile une librairie **libft.a**.
- Votre makefile doit afficher chaque commande qu'il effectue sans aucune fioriture.
- Votre makefile ne doit pas effectuer de commande inutile.
- Le **Makefile** ira chercher les fichiers sources dans le dossier **srcs**.
- Ces fichiers sources seront : **ft_putchar.c**, **ft_swap.c**, **ft_putstr.c**, **ft_strlen.c**, **ft_strcmp.c**
- Le **Makefile** ira chercher les fichiers headers dans le dossier **includes**.
- Ces fichiers headers seront : **ft.h**
- Il devra compiler vos fichiers **c** en utilisant **gcc** et les options **-Wall -Wextra -Werror** flags dans cet ordre.
- La lib sera à la racine de l'exercice.
- Les fichiers **.o** devront être à côté de leur fichier **.c** respectif.
- Le **Makefile** devra également implémenter des règles **clean**, **fclean**, **re**, la règle **all** et bien sur **libft.a**.
- Lancer juste **make** doit être équivalent à **make all**.
- La règle **all** devra faire comme **make libft.a**.
- la règle **clean** devra retirer tous les fichiers temporaires générés.
- La règle **fclean** fait l'équivalent d'un **make clean** et efface aussi le binaire créé

lors du `make all`.


- La règle `re` fait l'équivalent d'un `make fclean` puis un `make all`.
- Votre makefile ne doit pas recompiler les fichiers inutilement.
- Nous ne ramasserons que votre Makefile et testerons avec nos fichiers.



Attention aux wildcards!

Chapitre V

Exercice 02 : ft_split

	Exercice : 02
ft_split	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : ft_split.c	
Fonctions Autorisées : malloc	

- Écrire une fonction qui découpe une chaîne de caractères en fonction d'une autre chaîne de caractères.
- Il faudra utiliser chaque caractère de la chaîne **charset** comme séparateur.
- La fonction renvoie un tableau où chaque case contient l'adresse d'une chaîne de caractères comprise entre deux séparateurs. Le dernier élément 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);
```

Chapitre VI

Rendu et peer-evaluation

Rendez votre travail sur votre dépôt `Git` comme d'habitude. Seul le travail présent sur votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.

Vu que votre travail ne sera pas évalué par un programme, organisez vos fichiers comme bon vous semble du moment que vous rendez les fichiers obligatoires et respectez les consignes du sujet.



`Vous ne devez rendre uniquement les fichiers demandés par le sujet de ce projet.`