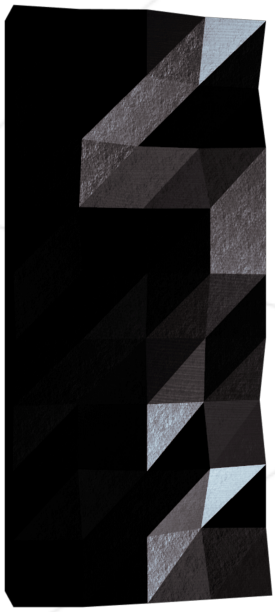

Piscine C

C 08

Staff 42 pedago@42.fr

Résumé: Ce document est le sujet du module C 08 de la piscine C de 42.

Table des matières

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 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

L'encyclopédie collaborative *Wikipédia* a ceci à dire sur l'ornithorynque :

L'ornithorynque (Ornithorhynchus anatinus) est une espèce de petits mammifères semi-aquatiques endémique de l'est de l'Australie, y compris la Tasmanie. C'est l'une des cinq espèces de l'ordre des monotrèmes, seul ordre de mammifères qui ponde des œufs au lieu de donner naissance à des petits complètement formés (les quatre autres espèces sont des échidnés). C'est la seule espèce survivante de la famille des Ornithorhynchidae et du genre Ornithorhynchus bien qu'un grand nombre de fragments d'espèces fossiles de cette famille et de ce genre aient été découverts.

L'apparence bizarre de ce mammifère pondant des œufs, muni d'aiguillons venimeux, à la mâchoire cornée ressemblant au bec d'un canard, à queue évoquant un castor, qui lui sert à la fois de gouvernail dans l'eau et de réserve de graisse, et à pattes de loutre a fortement surpris les premiers explorateurs qui l'ont découvert ; bon nombre de naturalistes européens ont cru à une plaisanterie. C'est l'un des rares mammifères venimeux : le mâle porte sur les pattes postérieures un aiguillon qui peut libérer du venin capable d'infliger de vives douleurs à un être humain. Les traits originaux de l'ornithorynque en font un sujet d'études important pour mieux comprendre l'évolution des espèces animales et en ont fait un des symboles de l'Australie : il a été utilisé comme mascotte pour de nombreux évènements nationaux et il figure au verso de la pièce de 20 cents australiens.

Jusqu'au début du XXe siècle, il a été chassé pour sa fourrure mais il est protégé à l'heure actuelle. Bien que les programmes de reproduction en captivité aient eu un succès très limité et qu'il soit sensible aux effets de la pollution, l'espèce n'est pas encore considérée comme en danger.

Ce sujet ne traite pas de l'ornithorynque.

Chapitre III

Exercice 00 : ft.h

	Exercice : 00
ft.h	
Dossier de rendu : $ex00/$	
Fichiers à rendre : `ft.h`	
Fonctions Autorisées : `Aucune`	

- Écrire votre fichier `ft.h`
- Il contient tous les prototypages des fonctions :

```
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);
```

Chapitre IV

Exercice 01 : ft_boolean.h

	Exercice : 01
ft_boolean.h	
Dossier de rendu : *ex*01/	
Fichiers à rendre : `ft_boolean.h`	
Fonctions Autorisées : `Aucune`	

- Écrire un fichier `ft_boolean.h` qui fera compiler et fonctionner correctement le main suivant :

```c
#include "ft_boolean.h"

void        ft_putstr(char *str)
{
        while (*str)
                write(1, str++, 1);
}

t_bool        ft_is_even(int nbr)
{
        return ((EVEN(nbr)) ? TRUE : FALSE);
}

int                main(int argc, char **argv)
{
        (void)argv;
        if (ft_is_even(argc - 1) == TRUE)
                ft_putstr(EVEN_MSG);
        else
                ft_putstr(ODD_MSG);
        return (SUCCESS);
}
```

- Ce programme devra afficher

```
I have an even number of arguments.
```

- ou

```
I have an odd number of arguments.
```

- suivi d'un retour à la ligne, dans le cas adéquat.

Chapitre V

Exercice 02 : ft_abs.h

	Exercice : 02
ft_abs.h	
Dossier de rendu : $ex02/$	
Fichiers à rendre : `ft_abs.h`	
Fonctions Autorisées : `Aucune`	

- Écrire une macro `ABS` qui remplace son paramètre par sa valeur absolue :

```
#define ABS(Value)
```

Chapitre VI

Exercice 03 : ft_point.h

	Exercice : 03
ft_point.h	
Dossier de rendu : $ex03/$	
Fichiers à rendre : `ft_point.h`	
Fonctions Autorisées : `Aucune`	

- Écrire un fichier `ft_point.h` qui fera compiler le main suivant :

```
#include "ft_point.h"

void        set_point(t_point *point)
{
        point->x = 42;
        point->y = 21;
}

int                main(void)
{
        t_point                point;

        set_point(&point);
        return (0);
}
```

Chapitre VII

Exercice 04 : ft_strs_to_tab

	Exercice : 04
ft_strs_to_tab	
Dossier de rendu : *ex*04/	
Fichiers à rendre : `ft_strs_to_tab.c`	
Fonctions Autorisées : `malloc, free`	

- Ecrire une fonction qui prend en parametre un tableau de chaîne de caractères ainsi que la taille de ce tableau et renvoie un tableau de structure.
- Elle devra être prototypée de la façon suivante :

```
struct s_stock_str *ft_strs_to_tab(int ac, char **av);
```

- Elle doit transformer chaque element du tableau de chaîne de caractères en structure.
- La structure sera définie dans le fichier `ft_stock_str.h` comme suit :

```
typedef struct s_stock_str
{
  int  size;
  char *str;
  char *copy;
}            t_stock_str;
```

  - `size` étant la taille de la chaîne de caractères ;
  - `str` étant la chaîne de caractères ;
  - `copy` étant une copie de la chaîne de caractères ;
- Elle doit garder l'ordre des elements de av.
- Le tableau de structures devra être alloué et le dernièr element aura 0 pour valeur de `str`, ceci afin de signifier la fin du tableau.

- Si une erreur d'allocation arrive elle doit renvoyer un pointeur NULL.
- Nous testons votre fonction avec notre `ft_show_tab` (exercice suivant). Prenez les mesures nécessaires pour que cela fonctionne !

Chapitre VIII

Exercice 05 : ft_show_tab

	Exercice : 05
ft_show_tab	
Dossier de rendu : *ex*05/	
Fichiers à rendre : `ft_show_tab.c`	
Fonctions Autorisées : `write`	

- Écrire une fonction qui affiche le contenu d'un tableau créé par la fonction précédente.
- Elle devra être prototypée de la façon suivante :

```
void ft_show_tab(struct s_stock_str *par);
```

- La structure est la même que l'éxercice précédent et sera dans le fichier `ft_stock_str.h` que nous vous fournirons, :
- Pour chaque élément du tableau :
  - la chaîne de caractères suivi d'un retour à la ligne
  - la taille suivi d'un retour à la ligne
  - la copie de la chaîne de caractères (qui aura pu être modifiée) suivi d'un retour à la ligne
- Nous testons votre fonction avec notre `ft_strs_to_tab` (exercice précédent). Prenez les mesures nécessaires pour que cela fonctionne !