



Piscine C

Journée 24

Staff 42 [pedago@42.fr](mailto:pedago@42.fr)

*Résumé: THE FOLLOWING TAKES PLACE BETWEEN 6.00 A.M. AND 7.00 A.M.*

# Table des matières

<b>I</b>	<b>Consignes</b>	<b>2</b>
<b>II</b>	<b>Préambule</b>	<b>4</b>
<b>III</b>	<b>ft_door</b>	<b>5</b>

# Chapitre I

## Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- 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.
- L'utilisation d'une fonction interdite est un cas de triche. Toute triche est sanctionnée par la note de -42.
- Si `ft_putchar()` est une fonction autorisée, nous compilerons avec notre `ft_putchar.c`.
- Vous ne devrez rendre une fonction `main()` que si nous vous demandons un programme.
- La Moulinette compile avec les flags `-Wall -Wextra -Werror`.
- Si votre programme ne compile pas, vous aurez 0.
- Les exercices shell doivent s'exécuter avec `/bin/sh`.
- 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 !
- 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.

# Chapitre II

## Préambule

C'était un réveil classique pour Chloe O'Brian, directrice rebelle du CTU New York.

Dans un râle, elle s'étira et maudit son alarme, réglée comme tous les jours sur 6 :00AM. Comme à son habitude, elle décida de faire le point sur sa situation. Un mal de crâne insoutenable lui donna pour indice que la nuit d'hier était très arrosée. Confiante, elle ouvra les yeux pour se rendre compte de son erreur sur toute la ligne. Chloé était allongée par terre, la tête légèrement ensanglantée, dans l'ascenseur du CTU d'où un message strident d'alerte était diffusé en continu : « Windows Elevator 8 a planté ». « Depuis quand suis-je coincé ? » se demanda-t-elle en se relevant.

Lentement, la chaîne des événements lui revint : comment elle était arrivé hier soir tard au bureau, le retour fracassant de Nick Bauer, le réparateur de photocopieuse qui n'avait pas de tournevis, les explosions grotesques de ses agents, son envie pressante d'utiliser l'ascenseur, la chute de l'appareil entre deux étages et sa commotion brutale.

« Oui voilà, c'est ça. »

Chloe pouvait commencer à paniquer.

« Non, je ne vais pas perdre mes moyens, je me sens plutôt en sécurité dans une cage d'ascenseur déjà tombée au dernier sous-sol. »

Cependant, Chloe succomba à la panique.

« D'autant plus qu'il y a de l'air pour plusieurs heures, et qu'en ouvrant la trappe d'évacuation, je n'ai rien à craindre. »

Pour des raisons dramatiques, Chloe était obligée de paniquer.


« Au secours ! cria Chloe en tambourinant la porte coincée de l'ascenseur. Je suis piégée ! »

Oui voilà, c'était bien ça.

Qui pourra venir délivrer Chloe ? Nick Bauer - aide-le à ouvrir l'ascenseur.

# Chapitre III

## ft\_door

	Exercice : 12
ft_door.h ft_door.c	
Dossier de rendu : <i>ex12/</i>	
Fichiers à rendre : <b>ft_door.h</b> , <b>ft_door.c</b>	
Fonctions Autorisées : <b>write</b>	
Remarques : n/a	

- Ecrire le `ft_door.h` et corriger le `ft_door.c` suivant :

```
#include "ft_door.h"

ft_putstr(char *str)
{
    int i = 0;

    while (str[i])
        write(1, str, i)

}

ft_bool close_door(t_door *door)
{
    ft_putstr("Door closing...\n");
    state = CLOSE;
    return (TRUE);
}

void is_door_open(t_door door)
{
    ft_putstr("Door is open ?");
    return (door->state = OPEN);
}

ft_bool is_door_close(t_door* door)
{
    ft_putstr("Door is close ?") ;
}
```

- Voici un exemple de main et de sortie.

```
$> cat main.c
#include <stdlib.h>
#include "ft_door.h"

int main()
{
    t_door    door;

    open_door(&door);
    if (is_door_close(&door))
        open_door(&door);
    if (is_door_open(&door))
        close_door(&door);
    if (door.state == OPEN)
        close_door(&door);
    return (EXIT_SUCCESS);
}
$> ./ft_door | cat -e
Door opening...$
Door is close ?$
Door is open ?$
Door closing...$
$>
```