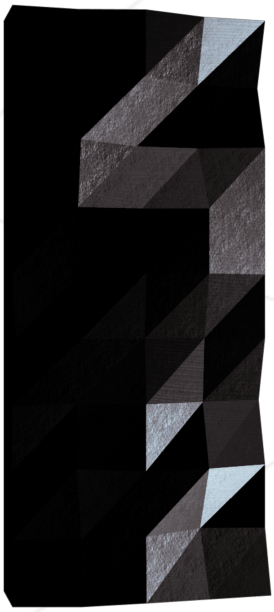

C Piscine

Day 24
The Spy Who Coded me

Staff 42 pedago@42.fr

Abstract: *THE FOLLOWING TAKES PLACE BETWEEN 3.00 A.M. AND 4.00 A.M.*

Contents

Chapter I

Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. So if you want to avoid bad surprises, be as thorough as possible.
- Moulinette is not very open-minded. It won't try and understand your code if it doesn't respect the Norm. Moulinette relies on a program called `Norminator` to check if your files respect the norm. TL;DR: it would be idiotic to submit a piece of work that doesn't pass `Norminator`'s check.
- These exercises are carefully laid out by order of difficulty - from easiest to hardest. We `will not` take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Using a forbidden function is considered cheating. Cheaters get `-42`, and this grade is non-negotiable.
- If ft_putchar() is an authorized function, we will compile your code with our `ft_putchar.c`.
- You'll only have to submit a main() function if we ask for a program.

- Moulinette compiles with these flags: -Wall -Wextra -Werror, and uses `gcc`.
- If your program doesn't compile, you'll get `0`.
- You cannot leave any additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / man / the Internet / ....`
- Check out the "C Piscine" part of the forum on the intranet.
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- By Odin, by Thor ! Use your brain !!!

Chapter II

Dr Evil V

« Austin Powers and Donnie Matrix are surrounded, Dr Evil, confirmed No 42. Should we arrest them ?
- Good answer, good answer ! Because what do we do when the intruders infiltrate our HQ ? Frau Farbissina, maybe ?
- We arrest THEM... ?
- 100% again, i can really feel this place is inspiring you ! Come we'll take the lift. »

We're going out of the board room - « Frau, switch off the lights, let's think a little about ecology of the planet that we'll destroy » – and let's go through what seems to be an empty field to get to the lift of my HQ. Ding.

« Ah, It's in the basement right now, let's wait a little shall we ? »

I hate to wait. Everybody is looking at me and expect me to start a conversation.

« So... Frau Farbissina... What do you think about the weather ?
- Hmm. It's nice ! »

At last, the lift stops at our level.

« Come on, come on, everybody in ! No 42, Frau, the one-legged clown, the dessert chariot, the pony, my decoy... Attention, i'm pressing the button 0! »

Grrrik. Grrrrriiiiikkkk.

Schboom.

« Dr Evil, i think we're stuck.
- Well done again No 42! We're too heavy, someone didn't go to the compulsory aerobic ! »

Inside myself, i cry.

Chapter III

ft_door

`Dr Evil is stuck in an elevator without conversation. Help him get out this terrible hell hole.`

	Exercice : 12
ft_door.h ft_door.c	
Turn-in directory : $ex12/$	
Files to turn in : `ft_door.h, ft_door.c`	
Allowed functions : `write`	
Remarks : `n/a`	

- Create the file ft_door.h and fix the following ft_door.c file :

```c
#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...");
        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 ?") ;
}
```

- Here's an example of main and output.

```
$> 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...$
$>
```