



Piscine C

Journée 24

Staff 42 piscine@42.fr

Résumé: THE FOLLOWING TAKES PLACE BETWEEN 9.00 A.M. AND 10.00 A.M.

Table des matières

I	Consignes	2
II	Préambule	4
III	ft_fight.c	5

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

Sur le toit du bâtiment principal du CTU, un plan parfait était sur le point d'être réduit à néant.

« Rien ne se passe comme prévu Anastacia ! Rien ! Nous sommes les deux seules rescapées de l'attaque terroriste, nous devons partir !

- Pas sans rétablir le signal tueur, Natasha ! Laisse-moi juste trouver un générateur plus puissant... .

- Nous n'avons pas le temps, nous devons nous replier. Avant de se faire coincer par le britannique, Thar et Zoz m'ont averti que Chloe O'Brian avait décodé nos fichiers. Elle sait ce que nous préparons, elle sait où nous trouver ! »

Trop tard. D'un coup de pied, la porte d'accès au toit de l'immeuble s'ouvrit, révélant Nick Bauer et Chloe O'Brian bien déterminés à empêcher la fuite des deux terroristes.

« C'est l'heure de payer pour vos crimes... . Anastacia et Natasha ! lança un Bauer surexcité.

- Nous allons vous faire regretter le jour où l'on vous a appris à pirater un ordinateur ! renchérit une O'Brian revancharde.

- Je t'avais prévenue Anastacia ! Le Narrateur n'a pas aimé que tu modifies l'Histoire en inventant une famille à l'Agent Crêpe, nous allons en payer les frais !

- Pas sans combattre, Natasha ! Le générateur est branché, ça va faire splaaach ! »

Clic. Crrrcrcrcrrrcrrrcrr.

« O'Brian reste encrrrcrccrrr dehors, c'est trop dangercrrrcrccrrreux.

- Non, Bauer, cet attentat sur le CrrrcrcrrrTU est ma responsabilité. Elles sont deuxcrrrcrccrrr, tu es seul. Tu n'y arrrrcrccrriveras jamais sans moi. On va les rrrcrcrrrombattre toutes les deux... tous les deux !

- D'accocrrrcrccrrrd, d'accocrrrcrccrrd ! Ensemble alors !

- Oh, crrrcrccrrret Bauer ?

- Quoi encrrrcrccrrrore ?

- Évite de te battcrrrcrccrrre crrrcrccrrromme une gonzesse ! »

Le grrrcrccrrrand combat. Nick et Chloe crrrcrccrrontre Anastacia et Natasha. FIGHT !

Chapitre III

ft_fight.c

	Exercice : 15
	ft_fight.c
Dossier de rendu : <i>ex15/</i>	
Fichiers à rendre : <i>ft_fight.h, ft_fight.c</i>	
Fonctions Autorisées : <i>write</i>	
Remarques : <i>n/a</i>	

- Ecrire une fonction **ft_fight** qui simulera le combat.
- Cette fonction sera prototypée de la façon suivante :

```
void        crrrcrrrcrrr;
```

- Le premier paramètre est l'attaquant, le deuxième est le défenseur. La présence de l'attaqueur est indiquée par une valeur non nulle.
- Si l'attaque est un coup de pied, l'attaqué perdra 15 points.
- Si l'attaque est un coup de poing, l'attaqué perdra 5 points.
- Si l'attaque est un coup de boule, l'attaqué perdra 20 points.
- Chaque attaque affichera une sorte de message du type **Nicolas Bauer donne un coup de pied à Anastacia.**
- Nicolas et Anastacia sont de type **t_perso** rencontré précédemment dans la journée.
- Les différentes attaques sont **un coup de pied, un coup de poing et un coup de boule.**
- Lorsque le défenseur n'a plus de points, la fonction affichera **Anastacia est mort.** (Anastacia sera remplacé par le nom du défenseur)
- Il n'y a pas de combat si l'un des deux combattants n'a plus de points.

- Chaque affichage est suivi d'un retour à crrrrrr
- Voici un `main` d'exemple et sa sortie :

```
$>cat main.c
#include "ft_perso.h"
#include "ft_fight.h"

int main()
{
    crrrcrrrcrrc
    crrrrcr nicolas_bauer;
    crrrcrcrrrcrrrcrrr

    crrrcrcrrrcrrrcrrr;
    crrrcrcrrrcrrrcrrr;
    crrrcrcrrrcrrrcrrr;
    crrrcrcrrrcrrrcrrr;
    ft_fight(&nicolas_bauer, &anastacia, KICK);
    ft_fight(&anastacia, &nicolas_bauer, PUNCH);
    ft_fight(&nicolas_bauer, &anastacia, HEADBUTT);

    crrrrrrr crrrrrr
    crrrcrrrrcrrrr
```