



# Piscine C

Mini-projet 01 : match / nmatch

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

*Résumé: Ce document est le sujet du mini-projet 01 de la piscine C de 42.  
Contrairement à ce qu'on pourrait croire, il ne contient aucune pastèque.*

# Table des matières

I	Préambule	2
II	Consignes	3
III	match	5
IV	nmatch	6

# Chapitre I

## Préambule

Voici ce que Wikipedia a à dire sur Shredder :

Shredder (Le Déchiqueteur en version québécoise), de son vrai nom Oroku Saki, est un personnage de la bande dessinée américaine des Tortues Ninja. Il est l'ennemi le plus acharné des Tortues Ninja et est inlassablement repris comme tel dans toutes les incarnations de la bande dessinée depuis ses débuts. Il s'agit aussi du chef de l'organisation connue comme le Clan des Foot.

Son apparence connaît des changements selon les versions, mais il est usuellement représenté comme un homme musclé, vêtu d'une tunique dont la couleur varie entre pourpre, rouge ou noir. Il porte une armure à peu près similaire dans toutes les versions, avec des brassards, épaulières et jambières couverts de lames et une sorte de casque de samourai surmonté de trois cornes formant une sorte de Trident et ne laissant que ses yeux visibles. Son arme habituelle, qu'il porte variablement sur ses deux mains ou sa main gauche (il est gaucher), est un gantelet surmonté de deux griffes, mais les lames sur son armure sont tout aussi mortelles que n'importe quelle arme.

L'idée de l'armure de Shredder est venue aux scénaristes en voyant une grande râpe à fromage triangulaire, les conduisant à imaginer un méchant qui aurait une arme de ce style sur les bras. Ils prévoyaient originellement de l'appeler le "Grate-Man", avant d'opter pour le nom plus sérieux de Shredder.

# Chapitre II

## 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. 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**.
- 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.
- 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.
- La Moulinette compile avec les flags `-Wall -Wextra -Werror`, et utilise `cc`.
- 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 autorisés.

ment 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.



Pour cette journée, la norminette doit être lancée avec le flag `-R CheckForbiddenSourceHeader`. La Moulinette l'utilisera aussi.

# Chapitre III

## match

	Exercice : 00
	match
Dossier de rendu :	<i>ex00/</i>
Fichiers à rendre :	<b>match.c</b>
Fonctions Autorisées :	Aucune
Remarques :	n/a

- Le but de la fonction est de savoir si deux chaînes matchent.
- On dit que **s1** et **s2** matchent lorsque **s1** et **s2** sont identiques.
- Si **s2** contient une étoile ('\*'), on pourra remplacer cette étoile par n'importe quelle chaîne de caractères (même vide) afin de rendre **s1** et **s2** identiques.
- **s2** peut contenir autant d'étoiles que l'on souhaite.
- Par exemple, "**main.c**" et "**\*.c**" matchent car il est possible de remplacer '\*' par la chaîne "**main**" pour faire en sorte que les deux chaînes soient identiques.
- Elle devra être prototypée de la façon suivante :

```
int      match(char *s1, char *s2);
```

- Elle devra renvoyer 1 si **s1** correspond à **s2** ou 0 dans le cas contraire.

# Chapitre IV

## nmatch

	Exercice : 01
	nmatch
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <b>nmatch.c</b>	
Fonctions Autorisées : Aucune	
Remarques : n/a	

- Le but de la fonction est de compter le nombre de fois que deux chaînes matchent.
- Lorsqu'on a deux étoiles ou plus, plusieurs combinaisons de chaînes de caractères peuvent convenir.
- **nmatch** calculera le nombre total de ces combinaisons.
- Quelques exemples :
  - "abcbd" et "\*b\*" matchent 2 fois : ("a", "cbd") et ("abc", "d")
  - "abc" et "a\*\*" matchent 3 fois : (rien, "bc"), ("b", "c") et ("bc", rien)
- Elle devra être prototypée de la façon suivante :

```
int      nmatch(char *s1, char *s2);
```

- Elle devra renvoyer le nombre de combinaisons qui matchent.