

# TD Machine #3 (5ième séance de TD) de CS101

- les 19/10/2022 et 21/10/2022
- Durée 1h30
- Pas de compte-rendu. Il est fortement recommandé de prendre des notes personnelles.
- Version 2022.1
- *author* Laure Gonnord

## À vos marques

- Le TDM est réalisé sur les machines de TP Linux de la salle B141.
- Essayer le plus possible de faire seul·e (on apprend moins vite en binôme)
- Ne pas hésiter à demander de l'aide (on apprend!)
- Créer un répertoire pour les TPs de C nommé CS101, et travailler dans un sous répertoire nommé TDM3.

## Partie 1: Petites fonctions simples

- Dans un nouveau fichier, rapidement créer une fonction main et faire en sorte que votre programme compile et exécute.

### à ce stade la ligne de commande de compilation doit être apprise/connue

- Écrire dans ce même fichier une fonction qui dessine des carrés d'étoiles de taille n, et une fonction qui dessine des triangles d'étoiles de taille n. (et les tester, bien sûr) Par exemple, pour n=3, cela donne:

```
***
***
***
```

et

```
*
* *
* * *
```

- Écrire dans ce même fichier une fonction qui prend en argument deux chaînes de caractères statiques de taille maximale N et qui retourne true si les chaînes sont identiques, false sinon. Tester.
- Dans le main, imprimer les codes ASCII des caractères 'a', 'A'. à quoi correspond la quantité 'A'-'a' ?
- Écrire dans ce même fichier une fonction qui "capitalise" un mot passé en paramètre en le copiant dans un deuxième tableau statique passé en paramètre aussi. De la même façon, on utilisera des chaînes statiques de taille maximale N. La fonction devra vérifier que chacun des caractères du tableau d'entrée comporte bien uniquement des caractères de l'alphabet latin (entre 'a' et 'z') On peut éventuellement utiliser la fonction isalpha de la bibliothèque string (et regarder la documentation avec `man isalpha`. Tester.

## Partie 2: Utilisation de code source fourni - lecture d'entiers dans un fichier.

- Récupérer sur Chamilo, répertoire code, le fichier nommé `tdm3.c`, le sauver dans votre répertoire TDM3, et vérifier que ce code compile et exécute correctement: la commande `./tdm3` doit imprimer un message d'erreur car le fichier `nombre` n'existe pas.
- Dans le même répertoire, créer un fichier `nombre` (sans extension) et mettre des nombres en colonne comme ceci:

```
12
7
9
25
```

Que fait l'exécution:

```
./tdm3
```

maintenant?

- Modifier ce programme (dans le main) pour:
  - compter le nombre de lignes du fichier et l'afficher à la fin de l'exécution.
  - calculer la moyenne des nombres du fichier.
  - remplir un tableau d'entiers de taille N fixé à l'avance (avec une constante symbolique). On arrêtera la lecture du fichier quand le tableau est plein.
- Refactorer le code: le main ne sera responsable que de l'ouverture du fichier et de coordonner le reste. Une fonction void `read_and_init(FILE *f, int t[N])` réalisera lecture et stockage dans un tableau, et une fonction `moyenne` calculera la moyenne d'un tableau.

### **Bonus : fonctions avancées de chaînes.**

- Écrire une fonction qui réalise la concaténation de deux chaînes contenues dans des tableaux statiques de taille au plus N. On effectuera cette concaténation dans un tableau statique de taille 2N.
- Écrire une fonction qui recherche une sous chaîne (tableau statique de taille N) dans une chaîne plus longue (taille 100N). Évaluer le nombre de comparaisons effectuées par votre fonction.