

TP3 : Passage de paramètres, pointeurs et structures

Séances 5 de TP de C – Evaluation à la discrétion du moniteur

[1. Objectifs](#)

[2. Les fonctionnalités attendues](#)

[Mode passage de paramètres](#)

[Arithmétique des pointeurs](#)

[3. Astuces](#)

[4. Evaluations](#)

1. Objectifs

L'objectif de cette séance est d'étudier, en langage C, les notions de passage de paramètre par copie de valeur, par adresse, et d'en apprécier les limites. La mise en oeuvre de structure de données nous permettra de lever les limites en terme de lisibilité. La notion de passage de paramètre par adresse se faisant par l'intermédiaire d'un pointeur, une partie de ce TP de C sera consacrée à l'arithmétique des pointeurs

Vous développerez cette bibliothèque dans des fichiers séparés afin de rendre votre code réutilisable. Vous devrez mettre en place des tests unitaires qui permettront de vérifier le bon comportement de vos fonctions.

L'utilisation de la fonction « scanf » est possible mais sera limitée aux nombres entiers ou flottants

2. Les fonctionnalités attendues

Ci-dessous est listé l'ensemble des fonctionnalités que devra fournir votre bibliothèque. Chacune de ces fonctionnalités possédera une fonction de test unitaire.

Le temps de codage est indiqué à titre informatif. Il correspond au temps qu'il devrait vous falloir pour coder chaque série de fonctionnalités, actuellement. Si vous mettez plus de temps, c'est que vous devez vous entraîner et pratiquer chez vous pour augmenter votre vitesse et ne plus être bloqué par des points de cours.

Mode de passages de paramètres

Ecrire une fonction qui recherche le nombre minimum d'un tableau d'entiers passés en paramètre :

- Solution attendue: une fonction qui renvoie le nombre minimum par return

En plus, de la recherche du nombre minimum, nous voulons aussi le nombre maximum et faire le calcul de la moyenne et de l'écart-type toujours dans la même fonction

- 1° solution attendue: une fonction avec plusieurs paramètres passés par adresse
- 2° solution attendue: une fonction avec une structure de données en paramètre passée par adresse qui devra contenir :
 - toutes les données en retour
 - et pourra aussi contenir les données en entrée (Tableau, taille, nombre d'éléments)

Ne pas oublier les tests unitaires. Bien choisir les fonctions de base.

▮ 30 minutes pour les fonctions + 30 minutes pour les tests unitaires

Arithmétique des pointeurs

- Reprendre la fonction `my_strcpy` du TP2 et travailler sur l'incrémentation des pointeurs. Ecrire un `my_strcpy2` qui copie 1 caractère sur 2 en travaillant sur l'incrémentation des pointeurs
- Reprendre la fonction `my_strlen` du TP2 et travailler sur différence entre 2 pointeurs pour trouver la longueur.
- Reprendre la fonction `my_strcat` et travailler sur l'ajout d'un entier à un pointeur, puis l'incrémentation (longueur rendue par `my_strlen` + `my_strcpy`).

Revalider les fonctions avec les tests unitaires déjà créées au TP2

▮ 20 minutes pour chaque exercice + 5 minutes pour les tests unitaires

3. Astuces

- Démarrez un nouveau projet – Ne repartez pas d'un ancien TP – Réfléchissez à l'organisation de vos fichiers sources
- Aidez-vous de vos TD d'algorithmique
- Réutilisez ce que vous avez fait au TP2 sur les fonctions de traitement de chaînes de caractères

- Concevez vos tests unitaires avant de coder vos fonctions correspondantes

4. Evaluations

Libre, laisser à la discrétion du moniteur