

TP2 : Chaines de caractères et tests unitaires

Séances 3 et 4 de TP de C – Evaluation du binôme en séance 4

[1. Objectifs](#)

[2. Les fonctionnalités attendues](#)

[1° partie : Etude du module « Seatest »](#)

[2° partie : Fonctions de traitements de chaîne de caractères](#)

[3° partie : Tests unitaires des fonctions](#)

[4° partie : Re-écritures des fonctions de traitements de chaînes de caractères](#)

[Bonus](#)

[3. Astuces](#)

[4. Evaluations](#)

1. Objectifs

L'objectif de cette séance est d'utiliser en langage C, les fonctions de traitements de chaînes de caractères.

Les notions de langages C qui vous seront utiles sont les tableaux, les fonctions de la bibliothèque « string.h » ainsi que les pointeurs sur tableaux de caractères

Vous devrez mettre en place des tests unitaires, à partir du module « [seatest](#) », qui permettront de vérifier le bon comportement de vos fonctions. Ces tests seront de nouveau utilisés lors de la ré-écriture de certaines fonctions.

L'utilisation de la fonction « scanf » est de nouveau interdite, même durant la phase de développement et de mise au point.

2. Les fonctionnalités attendues

Ci-dessous est listé l'ensemble des fonctionnalités que devra fournir votre TP de C.

Le temps de travail est indiqué à titre informatif. Il correspond au temps qu'il devrait vous falloir pour coder chaque partie.

1 ° partie : Etude du module « Seatest »

A partir de l'archive « seatest-V0.5.zip » fourni avec ce document, vous étudiez le module de tests unitaires « Seatest ». A partir de l'exemple fourni dans le dossier « exemples/nix », vous intégrerez les fichiers de librairie seatest.c & seatest.h dans un nouveau projet Netbeans.

□ 60 minutes

N'hésitez pas à modifier les exemples fournis. D'ailleurs, un test n'est pas concluant. Lequel et pourquoi ?

2 ° partie : Fonctions de traitements de chaînes de caractères

Dans un nouveau projet CodeBlocks, étudiez et mettez en œuvre des fonctions suivantes :

- isdigit()
- tolower() ou toupper()
- strlen()
- strcpy()
- strcat() & strncat()
- strcmp() & strncmp()

Proposer un jeu d'essais pour chaque fonction

□ 20 minutes pour chaque fonction

3 ° partie : Mise en œuvre des tests unitaires

Pour chaque fonction étudiée précédemment, mettez en œuvre les tests unitaires avec « Seatest », à partir des jeux d'essais que vous avez déterminé dans la partie 2.

Ne pas oublier d'intégrer les fichiers de la librairie dans votre projet.

- isdigit()
- tolower() (ou toupper())
- strlen()
- strcpy()
- strcat()
- strcmp()

□ 20 minutes pour chaque fonction

4° partie : Ré-écriture de fonctions

Voici l'ensemble des fonctions que vous aurez à réécrire :

- `my_isdigit()`
- `my_toupper()`
- `my_strlen()`
- `my_strcpy()`
- `my_strcat()`
- `my_strcmp()`

Vous vérifierez que les fonctions sont conformes en appliquant les tests unitaires de la partie 3 sur la fonction de référence, en vérifiant que les résultats sont identiques

▮ 30 minutes pour la mise en œuvre des fonctions

Bonus

Ecrire une fonction qui vérifie qu'un caractère est un caractère de ponctuation. Prévoir les jeux d'essais et les tests unitaires.

▮ 15 minutes

3. Astuces

- Démarrez toujours un nouveau projet – Ne dupliquez pas un ancien TP
- Etudiez bien les fonctions proposées en cherchant des exemples simples de mise en oeuvre
- Définissez vos jeux d'essais en même temps que la mise en œuvre des fonctions. Par exemple pour la fonction `isdigit()`, le jeu d'essais devra contenir des chiffres, mais aussi de la ponctuation et des lettres. Ainsi vous préparer la mise en œuvre des tests unitaires en même temps que le codage de vos fonctions.

4. Evaluations

- Maîtrise de CodeBlocks (Création d'un projet, ajout de fichiers sources c,h, compilation, debugger)
- Conventions de codage (commentaires, indentation, nom des fichiers, des fonctions, des variables, ...)
- Compilation séparée .h/.c, séparation du code en fonctions
- Tableaux, boucles, pointeurs sur caractères
- Pointeurs, passage de paramètres par valeur/par adresse
- Respect des contraintes de l'énoncé
- Choix des jeux d'essais
- Tests unitaires avec « Seatest »