

Travaux dirigés 6 Programmation en Langage C SMA3

Exercice 1 :

1. Écrire une fonction `strlen` qui calcule la longueur d'une chaîne.
2. Écrire une fonction `strcmp` qui compare deux chaînes de caractères pour l'ordre lexicographique. Cette fonction rendra -1, 0 ou 1 comme le fait la fonction `strcmp` de la librairie standard.
3. Écrire une fonction `delSpace` qui supprime les espaces et tabulations d'une chaîne de caractères. Cette fonction ne doit pas utiliser de tableau intermédiaire. Le morceau de code

```
char s[] = "bonjour à tous";  
delSpace(s);  
printf("s = \"%s\"\n", s);
```

doit afficher le résultat `s = "bonjouràtous"`.
4. Écrire une fonction `strsubs` qui substitue une chaîne de caractères dans une autre chaîne. Cette fonction prend les quatre paramètres suivants :
 - une chaîne *str* à l'intérieur de laquelle s'effectue la substitution.
 - une chaîne *motif* qui remplace une partie de la chaîne *str*.
 - une position *pos* dans la chaîne *str* à partir de laquelle on commence à remplacer les caractères de *str* par ceux de *motif*.
 - un nombre *len* qui donne le nombre maximal de caractères à substituer. La fonction en substitue moins si les fins des chaînes *str* ou *motif* sont atteintes avant.La fonction `strsubs` retourne le nombre de caractères qu'elle a effectivement substitués. Le morceau de code

```
char s[] = "bonjour à tous";  
strsubs(s, "soirée", 3, 4);  
printf("s = \"%s\"\n", s);
```

doit afficher le résultat `s = "bonsoir à tous"`.
5. Écrire une fonction `strrev` qui renverse une chaîne.
6. Écrire une fonction `strsort` qui trie par ordre alphabétique les caractères d'une chaîne. Écrire une première version avec des indices entiers puis écrire une nouvelle version qui utilise des pointeurs.
7. Réécrire la fonction `strcmp` en utilisant des pointeurs pour parcourir les chaînes au lieu d'entiers pour indexer les caractères.

Exercice 2 :

Ecrire la fonction **MonPrintf** qui simule le fonctionnement du **printf** pour les chaînes de caractères. Cette fonction prendra en paramètre une chaîne de caractères (sans "format" %) et l'affichera caractère par caractère.

Exercice 3 :

Vérifier si une suite de caractères tapée au clavier est correctement parenthésée (on doit refermer autant de parenthèses que de parenthèses ouvertes et jamais plus)

Exercice 4 :

Ecrire une fonction qui prend une chaîne de caractères *t* en paramètre et retourne cette même chaîne où les minuscules ont été transformées en majuscules.

Indice : une variable lettre contenant une minuscule peut être convertie en majuscule par la manipulation de son code

Exercice 5 :

1. Une chaîne de caractères est un palindrome si elle est identique à la chaîne retournée. Écrire programme qui vérifie sans utiliser une fonction de **<string>**, si une chaîne **CH** introduite au clavier est un palindrome en utilisant uniquement le formalisme tableau
2. Même question en utilisant des pointeurs au lieu des indices numériques

Exercice 6 :

1. Écrire une fonction qui prend comme argument deux chaînes de caractères **str** (une longue chaîne) et **word** (un mot), et qui renvoie le nombre de fois que **word** est présent dans **str**.
2. Écrire une fonction qui prend comme argument deux chaînes de caractères **str** (une longue chaîne) et **word** (un mot), et qui renvoie la première position de **word** dans **str** si **word** est présent dans **str**, et -1 sinon.