

TD 5 - Chaînes de caractères

Techniques de Programmation

Exercice 1

On utilisera le type : `typedef char mot[NMAX]` où `NMAX` est une constante entière préalablement définie. Pour cet exercice, vous n'utiliserez aucune des fonctions `strcpy`, `strcmp`, `strncmp`. Vous pourrez néanmoins utiliser `strlen`.

- Écrire et tester une fonction `miroir(mot dest, mot src)` rangeant dans un mot `dest` le miroir (mot à l'envers) du mot `src`.
- Écrire une fonction `void majuscules(mot dest, mot src)` convertissant les lettres minuscules du mot `src` en majuscules et en sauvegardant le résultat dans le mot `dest`.
- Écrire et tester :
 - une fonction `int est_facteur_gauche(mot facteur_g, mot src)` qui vérifie si `facteur_g` est facteur gauche de `src`. Par exemple, "poly" est facteur gauche de "polycopié".
 - une fonction `int est_facteur(mot facteur, mot src)` qui vérifie si le mot `facteur` est facteur de `src`. Par exemple, "poly", "copi", "polycop" sont facteurs de "polycopié".
 - une fonction `int est_sous_mot(mot sous_mot, mot src)` qui fait la même chose que `est_facteur`, mais de manière récursive.

Exercice 2

On utilisera les types suivants :

```
typedef char *mot; /* adresse d'une chaîne de caractères de longueur maximale MOTMAX=50*/  
typedef mot * langage; /* tableau de mots contenant au plus LANGMAX=100 éléments dont le dernier  
a pour valeur NULL */
```

- Écrire une fonction `void afficher(langage l, int iTaille)` affichant à l'écran les mots du langage `l`.
- Écrire une fonction `int appartient(mot m, langage l, int iNbMots)` retournant 1 ou 0 suivant que `m` est ou non un mot du langage `l`.
- Écrire une fonction `int disjoints(langage l1, langage l2, int iNbMot1, int iNbMot2)` retournant 1 ou 0 suivant que `l1` ou `l2` sont ou non disjoints.
- Écrire une fonction `int ajouter(mot m, langage l, int * iNbMots)` qui ajoute le mot `m` au langage `l` et retourne son indice dans le tableau (ou -1 si `m=NULL`).
- Écrire une fonction `void reunir(langage l1, langage *l2, int iNbMots1, int *iNbMots2)` qui ajoute au langage `l2` les mots de `l1` qui ne s'y trouvaient pas.