

Correction TD 2

Les chaînes de caractères

Exercices complémentaires

Exercice 3

Écrire les deux algorithmes de manipulation de chaînes de caractères suivants :

- *longueur*, qui prend en entrée une chaîne de caractères et qui renvoie sa longueur,
- *concaténation*, qui met bout à bout le contenu de deux chaînes de caractères et retourne le résultat,

Écrivez l'algorithme du programme principal qui demande 2 chaînes à l'utilisateur et affiche leur longueur et le résultat de leur concaténation.

```
ALGORITHME longueur
ENVIRONNEMENT
ENTREES
    chaine : tableau de n char
SORTIE
    entier # la longueur de la chaine (sans le '\0')
INTERNES
    index : entier

TRAITEMENT
DEBUT
    index ← 0
    TANT QUE chaine(index) ≠ '\0' FAIRE
        index ← index + 1
    FIN TANT QUE
    RETOURNER index
FIN
FIN ALGORITHME
```

```
ALGORITHME concatenation
ENVIRONNEMENT
ENTREES
    chaine_1 : tableau de n char
    chaine_2 : tableau de m char
    chaine_concat : tableau de n+m-1 char
SORTIE
    chaine_concat : tableau de n+m-1 char
INTERNES
    index : entier
    offset : entier

TRAITEMENT
DEBUT
    #copie chaine 1 dans chaine_concat
    POUR index ALLANT DE 0 A longueur(chaine_1) -1 PAR PAS DE 1
        chaine_concat(index) ← chaine_1(index)
    FIN POUR
    #copie chaine 2 dans chaine_concat
    offset ← longueur(chaine_1)
    POUR index ALLANT DE 0 A longueur(chaine_2) -1 PAR PAS DE 1
        chaine_concat(index + offset) ← chaine_2(index)
    FIN POUR
    #on ferme la chaine concaténée
    chaine_concat(longueur(chaine_1) + longueur(chaine_2) + 1 ) ← '\0'
    RETOURNER chaine_concat
FIN
FIN ALGORITHME
```

```

ALGORITHME programmeprincipal
  ENVIRONNEMENT
    chaine_A : tableau de n char
    chaine_B : tableau de m char
    chaine_C : tableau de n+m-1 char

  TRAITEMENT
    DEBUT
      ECRIRE "entrez chaine A"
      chaine_A ← saisir(chaine_A)
      ECRIRE "entrez chaine B"
      chaine_B ← saisir(chaine_B)
      ECRIRE longueur(chaine_A)
      ECRIRE longueur(chaine_B)
      chaine_C ← concatenation(chaine_A, chaine_B, chaine_C)
      afficher(chaine_C)
    FIN
FIN ALGORITHME

```

Exercice 4

Écrire un algorithme qui permet de comparer 2 chaînes de caractères : Elle retourne 0 si les deux chaînes de caractères sont identiques, -1 si la première chaîne est inférieure à la deuxième (ordre ASCII) et 1 sinon.

```

ALGORITHME minimum
  ENVIRONNEMENT
    ENTREES
      a : entier
      b : entier
    SORTIE
      entier # le plus petit des 2 entiers
    INTERNES
      néant

  TRAITEMENT
    DEBUT
      SI a < b FAIRE
        minimum ← a
      SINON
        minimum ← b
      FIN SI
    FIN
FIN ALGORITHME

```

```

ALGORITHME compare
  ENVIRONNEMENT
    ENTREES
      chaine_A : tableau de char
      chaine_B : tableau de char
    SORTIE
      entier # 0 si chaines égales, 1 si chaine_A > chaine_B, -1 sinon
    INTERNES
      i : entier

  TRAITEMENT
    DEBUT
      i ← 0
      TANT QUE chaine_A(i) = chaine_B(i) ET i < minimum(longueur(chaine_A), longueur(chaine_B))
        i ← i+1
      FIN TANT QUE
      SI (i > longueur(chaine_A)) ET (i > longueur(chaine_B)) FAIRE
        RETOURNER 0
      SINON SI chaine_A(i) > chaine_B(i) FAIRE
        RETOURNER 1
      SINON
        RETOURNER -1
      FIN SI
    FIN
FIN ALGORITHME

```

Exercice 5

1. Écrire une fonction qui prend en entrée un caractère, et qui retourne ce caractère après avoir changé sa casse. Le caractère n'est pas modifié s'il ne s'agit pas d'une lettre.

```
ALGORITHME changeCasse
ENVIRONNEMENT
  ENTREES
    c : char
  SORTIE
    char
  INTERNES
    néant

TRAITEMENT
  DEBUT
    SI (c ≥ 'a') ET (c ≤ 'z') FAIRE #si c'est une majuscule
      RETOURNER c + 32
    SINON SI (c ≥ 'A') ET (c ≤ 'Z') FAIRE #si c'est une minuscule
      RETOURNER c - 32
    SINON
      RETOURNER c #sinon inchangé
    FIN SI
    #ici on aurait pu utiliser la notation entière : SI (c ≥ 65) ET (c ≤ 90)
  FIN
FIN ALGORITHME
```

2. Écrire une fonction qui prend en entrée une chaîne de caractères, et qui affiche cette chaîne en inversant sa casse. Faites appel à cette fonction dans un programme principal.

```
ALGORITHME changeCasseChaine
ENVIRONNEMENT
  ENTREES
    chaine : tableau de char
  SORTIE
    tableau de char
  INTERNES
    index : entier

TRAITEMENT
  DEBUT
    POUR index ALLANT DE 0 A longueur(chaine) - 1
      chaine(i) ← changeCasse(chaine(i))
    FIN POUR
    RETOURNER chaine
  FIN
FIN ALGORITHME
```

```
ALGORITHME programme_principal
ENVIRONNEMENT
  chaine_A : tableau de 30 char

TRAITEMENT
  DEBUT
    ECRIRE "entrez un mot"
    chaine_A ← saisir(chaine_A)
    #on inverse la casse de la chaine
    chaine_A ← changeCasseChaine(chaine_A)
    ECRIRE "chaîne modifiée : ", afficher(chaine_A)
  FIN
FIN ALGORITHME
```