



# RÉVISION TABLEAUX ET CHAINES DE CARACTÈRES

AÏCHA EL GOLLI

[aicha.elgolli@essai.ucar.tn](mailto:aicha.elgolli@essai.ucar.tn)

Novembre-décembre 2022



ECOLE SUPÉRIEURE DE LA STATISTIQUE  
ET DE L'ANALYSE DE L'INFORMATION

On désire écrire un algorithme qui construit un tableau de caractères, calcul et affiche le nombre d'occurrences d'une la lettre et insère un caractère. L'algorithme, commenté par la suite, est le suivant :

### **Algorithme tableauAlea**

**variables** tabCarac [100] : **caractère** ;

nbVal : **entier** ;

**Début**

*{Procédure init qui initialise un tableau de n caractères est décrit ci-dessous}*

**init** (tabCarac, nbVal) ;

*{Appel à la fonction rechercher qui renvoie le nombre d'occurrences d'une lettre dans un tableau caractères et décrite ci-dessous}*

afficher ("le nombre d'occurrences de la lettre e est =: ", **rechercher** (tabCarac, nbVal, 'e')) ;

*{Procédure insert qui insère un caractère dans un tableau de n caractères est décrite ci-dessous}*

**insert** (tabCarac, nbVal) ;

**fin**

1. Définir une procédure **init** qui demande à l'utilisateur de saisir la valeur d'un entier ***n*** positif ou nul (vérifier que la valeur saisie est bonne, et redemander si nécessaire), et qui ensuite saisit les ***n*** éléments d'un tableau de caractères. N'oubliez pas que le tableau ainsi que sa taille doivent être des paramètres données/résultat.

Procédure init(**d/r** t []: caractère, **d/r** n : entier)

i : entier ;

Debut

Repeter

    Afficher("donner un nombre positif inférieur à 100 ") ;

    Entrer(n) ;

Jusqu'à (n>=0)&&(n<=100) ;

Pour(i de 0 à n-1) faire

    Entrer(t[i]) ;

fin

2. Définir une fonction **rechercher** qui reçoit un tableau de *n caractères* en paramètre ainsi que la lettre à rechercher, et qui calcule et renvoie le nombre d'occurrences d'une lettre donnée dans le tableau.

Fonction **rechercher**(t[] : caractère, n : entier, c : caractère) : **entier**

Occ, i : entier

Debut

Occ  $\leftarrow$  0 ;

Pour(i de 0 à n-1) faire

    Si t[i]=c alors occ  $\leftarrow$  occ+1 ;

Retourne occ ;

fin

3. Définir une procédure **insert** qui permet d'insérer un caractère dans un tableau de caractères. La procédure **insert** reçoit en paramètre un tableau **T** de caractères et l'entier **n** représentant la taille réelle du tableau T. Cette procédure **insert** demande à l'utilisateur le caractère à insérer et la position dans laquelle sera insérer le caractère (cette position doit être comprise entre **0** et **n**).

Procédure insert(**d/r** t []: caractère, **d/r** n : entier)

c : caractere ;

pos,j: entier

Debut

Si (n<100)

Afficher("entrer le caractere a inserer ") ;

Entrer(c) ;

Repeter

Entrer(pos)

Jusqu'à (pos>=0)et((pos<n) ;

Pour(j=n-1 à pos){ pas -1 } faire

    T[j+1]←t[j] ; FinPour;

T[pos]←c ;

n=n+1 ;

Finsi ;

Fin ;

4. Dites ce que fait cette procédure?

Procédure Mystere(**d/r** t [ ]:entier, **d/r** n : entier)

i, j : entier ;

Début

    j ← 0;

    Pour(i ← 0 à n-1)faire

        t[j] ← t[i];

        Si(t[i] <> 0) alors     j ← j+1;

    FPour ;

    n ← j;

Fin ;

Ecrire un programme qui lit une ligne de texte (ne dépassant pas 200 caractères) la mémorise dans une variable TXT et affiche ensuite:

- a) la longueur L de la chaîne.
- b) le nombre de 'e' contenus dans le texte.
- c) toute la phrase à rebours, sans changer le contenu de la variable TXT.
- d) toute la phrase à rebours, après avoir inversé l'ordre des caractères dans TXT:

voici une petite phrase ! ! esarhp etitep enu iciov

```

#include <stdio.h>
main()
{
    /* Déclarations */
    char TXT[201]; /* chaîne donnée */
    int I,J; /* indices courants */
    int L; /* longueur de la chaîne */
    int C; /* compteur des lettres 'e' */
    int AIDE; /* pour l'échange des caractères */

    /* Saisie des données */
    printf("Entrez une ligne de texte (max.200 caractères):\n");
    gets(TXT); /* L'utilisation de scanf est impossible pour */
    /* lire une phrase contenant un nombre variable de mots. */

    /* a) Compter les caractères */
    /* La marque de fin de chaîne '\0' est */
    /* utilisée comme condition d'arrêt. */
    for (L=0; TXT[L]; L++);
    printf("Le texte est composé de %d caractères.\n",L);

    /* b) Compter les lettres 'e' dans le texte */
    C=0;
    for (I=0; TXT[I]; I++)
        if (TXT[I]=='e') C++;
    printf("Le texte contient %d lettres \\'e\\'.\n",C);

```

```

    /* c) Afficher la phrase à l'envers */
    for (I=L-1; I>=0; I--)
        putchar(TXT[I]); /* ou printf("%c",TXT[I]); */
    putchar('\n'); /* ou printf("\n"); */

    /* d) Inverser l'ordre des caractères */
    for (I=0,J=L-1 ; I<J ; I++,J--)
    {
        AIDE=TXT[I];
        TXT[I]=TXT[J];
        TXT[J]=AIDE;
    }
    puts(TXT); /* ou printf("%s\n",TXT); */
    return 0;
}

```