

Correction TD8 ASD : Les chaînes de caractères

Exercice 1

Procédure SAISIE_CH(CH : chaîne)

DÉBUT

Répéter

Écrire ("Donner une chaîne :")

Lire (CH)

Jusqu'à (Long (CH) \neq 0)

FIN

Fonction TEST (c : caractère ; CH : chaîne) : Booléen

VAR

i : entier T : Booléen

DÉBUT

i \leftarrow 1 T \leftarrow faux

Tantque (i \leq Long (CH) et T = faux) faire

Si (CH[i] = c) alors

T \leftarrow vrai

Sinon

i \leftarrow i + 1

Finsi

Fintantque

TEST \leftarrow T

FIN

Procédure PRESENTS (CH1, CH2 , CH3 : chaîne)

VAR i, j : entier

DÉBUT

j \leftarrow 1

Pour i de 1 à Long (CH1) faire

Si ((TEST(CH1[i] , CH2) = vrai) ET (TEST (CH1[i] , CH3) = faux)) alors

CH3[j] \leftarrow CH1[i]

j \leftarrow j+ 1

Finsi

Fin Pour

Écrire ("La chaîne CH3 est :", CH3)

FIN

Procédure ABSENTS (CH1, CH2 , CH4 : chaîne)

VAR

i, j : entier

DÉBUT

j \leftarrow 1

Pour i de 1 à Long (CH1) faire

Si ((TEST(CH1[i] , CH2) = faux) ET (TEST (CH1[i] , CH3) = faux)) alors

CH4[j] \leftarrow CH1[i]

j \leftarrow j+ 1

Finsi

Fin Pour

Écrire ("La chaîne CH4 est :", CH4)

FIN

Procédure ALTERNER (CH1, CH2 , CH5 : chaîne)

VAR

i, j, k : entier

DÉBUT

i ← 1

j ← 1

k ← 1

Tant que (i ≤ Long(CH1) et (j ≤ Long(CH2)) Faire

CH5[k] ← CH1[i]

k ← k + 1

CH5[k] ← CH2[j]

k ← k + 1

i ← i + 1 j ← j + 1

Fin Tant que

Tant que (i ≤ Long(CH1)) Faire

CH5[k] ← CH1[i];

k ← k + 1

i ← i + 1

Fin Tant que

Tant que (j ≤ Long(CH2)) Faire

CH5[k] ← CH2[j];

k ← k + 1

j ← j + 1

Fin Tant que

Écrire ("La chaîne alternée est : ", CH5)

FIN

Procédure CRYPTER (CH2 : chaîne)

VAR

i, j : entier

caux : caractère

DÉBUT

i ← 1 j ← 2

Tant que (i ≤ Long(CH2) et (j ≤ Long(CH2)) Faire

caux ← CH2[i]

CH2[i] ← CH2[i+1]

CH2[i+1] ← caux

Fin Tant que

Écrire ("La chaîne crypter est : ", CH2)

FIN

Procédure DECALE (CH1 : chaîne[30])

VAR

i : entier

x : caractère

DÉBUT

x ← CH1[LONG(CH1)]

Pour i de LONG(CH1) à 2 (pas = -1) Faire

CH1[i] ← CH1[i - 1]

Fin Pour

CH1[i] ← x

FIN

Procédure ROT_CIRC (CH1 : chaîne[30] ; K : entier)

VAR

i : entier

DÉBUT

Pour i de 1 à K Faire

DECALE (CH1)

Fin Pour

Écrire ("La chaîne résultat est : ", CH1)

FIN

Algorithme exercice1

VAR

CH1, CH2, CH3, CH4, CH5 : chaîne

K : entier

DÉBUT

SAISIE_CH (CH1)

SAISIE_CH (CH2)

Répéter

Écrire ("donner un entier :")

Lire (K)

Jusqu'à (K <= LONG(CH1))

PRESENTS (CH1, CH2, CH3)

ABSENTS (CH1, CH2, CH3)

ALTERNER(CH1, CH2, CH5)

ROT_CIRC (CH1, K)

CRYPTER(CH2)

FIN

Exercice 2

Procédure Remplir (t : tab ; n : entier)

VAR

ientier

DÉBUT

Répéter

Écrire ("n=")

Lire(n)

Jusqu'à (n>=2) ET (n<=30)

Pour i de 1 à n Faire

Répéter

Écrire ("T[" , i , "]=")

Lire(t[i])

Jusqu'à (Long(t[i])>0) Et (Long (t[i])<=5)

Fin Pour

FIN

Procédure Eliminer(t: tab ; n: entier)

VAR

j , i : entier

ch : chaîne

DÉBUT

Pour i de 1 à n Faire

ch←""

Pour j de 1 à LONG(t[i]) Faire

Si ((t[i][j] <='a' et t[i][j] <= 'z') OU (t[i][j] <='A' et t[i][j] <= 'Z')) Alors

ch ← ch + t[i][j]

```

        Fin Si
        t[i] ← ch
    Fin Pour
Fin Pour
FIN
Procédure majus(t : tab ; n : entier )
VAR      i , j : entier
DÉBUT
    Pour i de 1 à n Faire
        Pour j de 1 à Long(t[i]) Faire
            t[i][j] ← MAJUS(t[i][j])
        Fin Pour
    Fin Pour
FIN
Procédure affichage(t : tab ; n : entier)
VAR      i , j, L : entier
           palindrome : booléen
DÉBUT
    Pour i de 1 à n faire
        L ← Long(t[i])
        palindrome ← vrai
        Pour j de 1 to L div 2 faire
            Si (t[i][j] < > t[i][L-j+1]) Alors
                palindrome ← Faux
            Finsi
        Fin Pour
        Si (palindrome) Alors
            Écrire ("    ", t[i])
        Finsi
    Fin Pour
FIN

```

Algorithme Exercice 2

```

TYPE      tab= Tableau [1..50][1..5] de caractères
VAR      ttab
           n:entier
DÉBUT
    remplir(t,n)
    eliminer(t,n)
    majus(t,n)
    affichage(t,n)

```

FIN

Exercice 3

Procédure SAISIE (m , ph : chaîne)

```

VAR      test : booléen
           i , l : entier

```

```

DÉBUT
    Répéter

```

```

    Écrire ("saisir un mot : ")
    Lire(m)
    test ← vrai
    i ← 0
    l ← Long(m)
    Tant que (test) ET (i<l) Faire
        i ← i+1
        test ← MAJUS(m[i]) <='A' et MAJUS(m[i])<= 'Z']
    Fin Tant que
    Jusqu'à test
    Écrire ("saisir une phrase : ")
    Lire (ph)

```

FIN

Procédure CONVERTIR (ch : chaîne)

VAR

i : entier

DÉBUT

```

    Pour i de 1 à Long(ch) Faire
        ch[i] ← MAJUS(t[i][j])
    Fin Pour

```

FIN

Fonction FREQUENCE(m, ph: chaîne) : entier

VAR

ph1: chaîne
f : entier

DÉBUT

```

    f←0
    ph1← ph
    Répéter
        Si pos(m,ph1)< >0 Alors
            f←f+1
            EFFACER(ph1,pos(m,ph1),LONG(m))
        Fin Si
    Jusqu'à pos(m,ph1) = 0
    FREQUENCE←f

```

FIN

Algorithme Exercice 3

VAR

m , ph : chaîne

DÉBUT

```

    SAISIE(m,ph)
    CONVERTIR(m)
    CONVERTIR (ph)
    Écrire ("le nombre d'apparition est : ", FREQUENCE(m, ph))

```

FIN