Université Ibn Zohr
Faculté des Sciences d'Agadir
Département d'Informatique
A.U : 2016 / 2017

Proposition de Correction de l'Examen de Session de Rattrapage d'Algorithmique 2
Filière : SMI3
Durée : 2 h - Documents non autorisés

Exercice 1 : [1.5 points][~10 minutes]

Ecrire une fonction récursive **MULT4** qui prend en paramètre un entier n et qui retourne 1 si n est multiple de 4 et 0 sinon, et ce **sans utiliser la fonction mod ou l'opérateur %**

```
Fonction MULT4(n :Entier) : Entier
```

Solution :

```
Fonction MULT4(n :Entier) : Entier
variable pc : ^ caractère
Début
Si n = 0
 retourner 1
 Sinon si n = 1 OU n = 2 ou n = 3
       retourner 0
       Sinon retourner (MULT4(n-4))
Fin
```

Exercice 2 : chaines de caractères [6.5 points][~35 minutes]

Rappel :

- **Une chaine de caractères est un tableau de caractères qui se termine par le caractère '\0'**
- **Dans cet exercice, vous pouvez utiliser les fonctions définies dans des questions précédentes.**

[1 pt]

1. Ecrire une fonction **INV()** qui retourne 1 si une chaine CH1 est l'inverse d'une chaine CH2 et retourne 0 sinon. On suppose que les deux chaines sont de même longueur L.
   Ex : CH1 = "ABCD" et CH2 = "DCBA" alors INV(CH1,CH2) retourne 1.

```
Fonction INV(CH1[] : caractère, CH2[] : caractère, L :Entier) : Entier
```

Solution :

```
Fonction INV(CH1[] : caractère, CH2[] : caractère, L :Entier) : Entier
Variable i : Entier
Début
Pour i allant de 0 à L-1 faire
 Si CH1[i] <> CH2[L-1-i] alors
 Retourner 0
 FinSi
FinPour
Retourner 1
Fin
```

[1 pt]

2. Ecrire une fonction récursive **IDR()** qui compare deux chaines de caractère et retourne 1 si les deux chaines sont identiques, sinon elle retourne 0.

```
Fonction IDR(CH1[] : caractère, CH2[] : caractère) : Entier
```

Solution :

```
Fonction IDR(CH1[] : caractère, CH2[] : caractère) : Entier
Début
Si CH1[0]='\0' ET CH2[0]='\0'
   retourner 1
Sinon si CH1[0] <> CH2[0]
        retourner 0
        Sinon retourner IDR(CH1 + 1,CH2 + 1)
        FinSi
FinSi
Fin
```

Un mot M2 est anagramme à un mot M1, si M2 est obtenu après un réarrangement des lettres du mot M1. Les mots M1 et M2 doivent avoir exactement les mêmes lettres et par conséquent la même longueur.
Exemple : M1 = "argent" et M2 = "tanger"

Université Ibn Zohr
Faculté des Sciences d'Agadir
Département d'Informatique
A.U : 2016 / 2017

[1 pt]

3. Ecrire la fonction **pos_LM()** qui retourne la position de la 1ère occurrence d'un caractère **car** dans une chaine de caractère **CH**. Si le caractère ne figure pas dans la chaine **CH** la fonction retourne -1.

```
Fontion pos_LM(ch[] : caractère, car : caractère) : Entier
```

Solution :

```
Fontion pos_LM(ch[] : caractère, car : caractère) : Entier
Variable i : Entier
Début
i ← 0
TantQue (ch[i] <> '\0')
   si (ch[i] = car)
   Retourner i
   FinSi
   i ← i+1
FinTantQue
Retourner -1
Fin
```

[1 pt]

4. Ecrire une fonction **supp_LM()** qui supprime une lettre **car** d'une chaine **CH**. La longueur de la chaine **CH** diminue de 1 après l'appel à la fonction **supp_LM().** S'il y a plusieurs occurrences de la lettre **car** dans **CH**, on supprime la 1ère.

Ex : CH="preference" et car='e' alors après supp_LM(CH,car), CH est "prference"

N.B. On exclue la possibilité que la lettre **car** n'appartient pas à la chaine **CH**.

```
Fonction supp_LM(CH[] :caractère, car : caractère) : vide
```

Solution :

```
Fonction supp_LM(CH[] :caractère, car : caractère) : vide
Variable i : Entier
Début
i ← 0
TantQue (ch[i] <> car)
   i ← i+1
FinTantQue
TantQue (ch[i] <> '\0')
   ch[i] ← ch[i+1]
   i ← i+1
FinTantQue
Fin
```

[2.5 pts]

5. Ecrire une fonction **ANAG()** qui retourne 1 si une chaîne de caractères CH1 est anagramme à une chaîne de caractères CH2, sinon elle retourne 0. La fonction **ANAG()** ne doit pas changer le contenu des chaines CH1 et CH2. On suppose que CH1 et CH2 **ont la même longueur**.

indication : Vous pouvez utiliser la fonction **CPY(CH1,CH2)** qui copie la chaine CH2 dans CH1

```
Fonction ANAG(CH1[] : caractère, CH2[] : caractère) : Entier
```

Solution :

```
Fonction ANAG(CH1[] : caractère, CH2[] : caractère) : Entier
Variable i : Entier
         CHA[31],CHB[31] : caractère
Début
CPY(CHA,CH1)
CPY(CHB,CH2)
i ← 0
TantQue (CHA[i] <> '\0')
   Si (pos_LM(CHB, CHA[i]) = -1)  /* CHA[i] n'appartient pas à CHB */
   Retourner 0
   Sinon
   supp_LM(CHB, CHA[i])
   i ← i+1
FinTantQue
Retourner 1
Fin
```

Université Ibn Zohr
Faculté des Sciences d'Agadir
Département d'Informatique
A.U : 2016 / 2017

Exercice 3 : Traitement des matrices à coefficients dans $\mathbb{Z}$. **[8 points][~55 minutes]**

[1.5 pt] 1. **Une matrice triangulaire supérieure** est une matrice carrée dont toutes les valeurs au-dessous de la diagonale principale sont nulles.

Ex : $\begin{pmatrix} 5 & 3 & 2 & 8 & 10 \\ 0 & -9 & 12 & 1 & 6 \\ 0 & 0 & 7 & 3 & 7 \\ 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}$

Ecrire une fonction **`TRSUP()`** qui retourne 1 si une matrice carrée A de taille nxn est triangulaire supérieure, sinon retourne 0. L'entête de la fonction est :

`Fonction TRSUP(A[ ][ ] : Entier, n : Entier) : Entier`

Solution :

```
Fonction TRSUP(A[ ][ ] : Entier, n : Entier) : Entier
Variable i : Entier
Début
Pour i allant de 1 à n-1
Pour j allant de 0 à i-1
 Si A[i][j] <> 0
 Retourner 0
 FinSi
FinPour // Pour j
FinPour // pour i

Retourner 1
Fin
```

[1.5 pt] 2. **Une matrice triangulaire inférieure** est une matrice carrée dont les valeurs au-dessus de la diagonale principale sont nulles.

Ex : $\begin{pmatrix} 5 & 0 & 0 & 0 & 0 \\ 7 & -9 & 0 & 0 & 0 \\ -1 & 2 & 7 & 0 & 0 \\ 11 & 4 & 3 & 2 & 0 \\ 6 & 3 & 1 & 7 & 3 \end{pmatrix}$

Ecrire une fonction **`TRINF()`** qui retourne 1 si une matrice carrée B de taille nxn est triangulaire inférieure, sinon retourne 0. L'entête de la fonction est :

`Fonction TRINF(B[ ][ ] : Entier, n : Entier) : Entier`

Solution :

```
Fonction TRINF (B[ ][ ] : Entier, n : Entier) : Entier
Variable i : Entier
Début
Pour i allant de 0 à n-2
Pour j allant de i+1 à n-1
 Si B[i][j] <> 0
 Retourner 0
 FinSi
FinPour // Pour j
FinPour // pour i

Retourner 1
Fin
```

[1 pt] 3. Une matrice triangulaire est une matrice triangulaire supérieure ou inférieure.
A partir des questions 1 et 2, en déduire une fonction **`Triang`** () qui retourne 1 si une matrice carrée T de taille nxn est triangulaire, sinon retourne 0. L'entête de la fonction est :

`Fonction Triang(T[ ][ ] : Entier, n : Entier) : Entier`

Solution :

```
Fonction Triang(T[ ][ ] : Entier, n : Entier) : Entier
Début
Si (TRSUP(T,n) = 1 OU TRINF(T,n) = 1)
Retourner 1
FinSi
retourner 0
Fin
```

[1 pt] 4. Une **matrice diagonale** est une matrice carrée dont les coefficients en dehors de la diagonale principale sont nuls.

Ex : $\begin{pmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & -9 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}$

A partir des questions 1 et 2, en déduire une fonction qui retourne 1 si une matrice carrée D de taille nxn est diagonale, sinon retourne 0. L'entête de la fonction est :
`Fonction Diag(D[ ][ ] : Entier, n : Entier) : Entier`

Solution :

```
Fonction Diag(D[ ][ ] : Entier, n : Entier) : Entier
Début
Si (TRSUP(T,n) = 1 ET TRINF(T,n) = 1)
Retourner 1
FinSi
retourner 0
Fin
```

[3 pts] 5. Une matrice de Toeplitz est une matrice carrée dont les coefficients sur chaque diagonale descendant de gauche à droite sont les mêmes.

Ex : $\begin{pmatrix} 1 & 7 & 9 & 4 & 3 \\ 13 & 1 & 7 & 9 & 4 \\ 8 & 13 & 1 & 7 & 9 \\ 6 & 8 & 13 & 1 & 7 \\ 14 & 6 & 8 & 13 & 1 \end{pmatrix}$

Ecrire une fonction `TOEP()` qui retourne 1 si une matrice carrée M de taille nxn est de Toeplitz, sinon retourne 0. L'entête de la fonction est :
`Fonction TOEP(M[ ][ ] : Entier, n : Entier) : Entier`

Solution :

```
Fonction TOEP(M[ ][ ] : Entier, n : Entier) : Entier
variables i,j: Entier
Début

Pour i allant de 0 à n-2
 Pour j allant de 0 à n-2
 Si (M[i][j] <> M[i+1][j+1]) alors
 retourner 0
 Finsi
 FinPour
FinPour

retourner 1
Fin
```

Université Ibn Zohr
Faculté des Sciences d'Agadir
Département d'Informatique
A.U : 2016 / 2017

<u>Exercice 4 :</u> Récursivité [4 points][~20 minutes]

Soit la fonction récursive **T** définie comme suit :

$$T(x,y,z)=\begin{cases} T(T(x-1,y,z),T(y-1,z,x),T(z-1,x,y)) & si \quad y<x \\ y & si \quad non \end{cases}$$

[2 pts] 1. Calculer en justifiant votre réponse les valeurs suivantes :

- `T(1,0,2) = T[T(0,0,2),T(-1,2,1),T(1,1,0)] = T(0,2,1) = 2`
- `T(1,0,1) = T[T(0,0,1),T(-1,1,1),T(0,1,0)] = T(0,1,1) = 1`

[1.5 pt] 2. Utiliser la question 1 pour calculer les valeurs suivantes : (justifier votre réponse)

- `T(2,1,0) = T[T(1,1,0),T(0,0,2),T(-1,2,1)] = T(1,0,2) = 2`
- `T(2,0,1) = T[T(1,0,1),T(-1,1,2),T(0,2,0)] = T(1,1,2) = 1`
- `T(1,5,0) = 5`

[0.5 pt] 3. A partir de la question 2, en déduire une version itérative de la fonction T.

```
Fonction T_iter(x : Entier, y : Entier, z : Entier) : Entier
Début
Si x ≤ y alors
 Retourner y
Sinon si y ≤ z alors
      Retourner z
      Sinon retourner x
      FinSi
FinSi
Fin
```