

TD 02 : Pointeurs & Fonctions

Partie Pointeurs

Exercice 1:

Proposer un algorithme qui lit une chaîne de caractères **ch** et affiche la longueur de la chaîne à l'aide d'un pointeur **pc**. Le programme n'utilisera pas de variables numériques.

indication : une chaîne de caractères est un tableau de caractère qui se termine par le premier caractère '\0'

Exercice 2:

Écrire un algorithme qui a en entrée une chaîne de caractères **ch** de longueur **L** et affiche si **ch** est palindrome ou pas.

Le parcours de la chaîne de caractères se fera via des pointeurs.

NB. Une chaîne de caractères est palindrome si elle se lit de la même manière de la droite et de la gauche.

Ex : "radar", "1221",...

Exercice 3:

Soit **T** un tableau d'entiers de taille **N**. Écrire un algorithme qui range les éléments pairs du tableau **T** en premiers et les éléments impairs en dernier.

La solution proposée doit impérativement utiliser les pointeurs uniquement. (Le programme n'utilisera pas de variables numériques à l'exception de **N**)

Remarque : le tableau n'est pas nécessairement trié.

Exemple :

Avant l'application de l'algorithme : **T[] = {1,8,3,4,9}**

Après l'application de l'algorithme : **T[] = {4,8,9,3,1}**

Partie Fonctions :

NB. Une procédure est une fonction qui ne retourne rien.

Exercice 4:

- Écrire une fonction **prod_1()** qui prend en paramètre deux entiers et retourne leur produit.
- Écrire la procédure **prod_2()** qui prend en paramètre, deux entiers et un troisième paramètre qui contient le résultat du produit.

Exercice 5:

Écrire une fonction **moy()** qui prend en paramètre un tableau d'entier **Tab** et la taille du tableau, et qui retourne la moyenne des éléments du tableau **Tab**.

Exercice 6:

Écrire une fonction **MinMax()** qui détermine quelles sont les valeurs **Min** et **Max** d'un tableau d'entiers **T** de taille **N**. La fonction **MinMax** ne doit rien afficher.

Exercice 7:

Soit **T** un tableau d'entier de taille **N**. Écrire une fonction qui copie toutes les composantes strictement positives de **T** dans un tableau **TPOS** et toutes les valeurs strictement négatives dans un tableau **TNEG**.

Indication : les tableaux **TPOS** et **TNEG** sont passés comme paramètres de la fonction.

Exercice 8 : [4 points][~ 20 minutes]

Un nombre premier est un entier naturel qui admet exactement deux diviseurs distincts entiers et positifs, qui sont 1 et lui-même.

Exemples : 19 ; 31 ; 61

1. Écrire une fonction qui prend en paramètre un entier naturel **P** et retourne 1 si **P** est un nombre premier et 0 sinon.

- Écrire une fonction **SPI** qui prend en paramètre un entier naturel N et retourne la Somme des nombres Premiers Inférieurs ou égal à N.
 Exemples : si N= 6 alors $SPI(6) = 2 + 3 + 5 = 10$
 si N = 7 alors $SPI(7) = 2 + 3 + 5 + 7 = 17$
- Un nombre K est P-premier, si K est premier **ET** si SPI(K) est premier.
 Exemple : 7 est P-premier.
 Écrire une fonction Ppremier qui prend en paramètre un entier naturel K et retourne 1 si K est P-premier et 0 sinon.
- En justifiant votre réponse, calculer si les entiers suivants sont P-premiers ou pas :
 - N = 13 ; N = 16 ; N = 19 ; N = 27 ;

Exercice 9 : (Expression bien parenthésée)

Une expression est dite bien parenthésée si :

- Pour toute parenthèse fermante, il existe une parenthèse ouvrante qui la précède, et qui n'a pas encore été fermée.
- Toute parenthèse ouvrante est fermée par une parenthèse fermante.

Écrire une fonction ayant en paramètre une chaîne de caractère ch (tableau de caractère), sa longueur N et retourne 1 si la chaîne ch est bien parenthésée et 0 sinon.

Exemples :

Les expressions suivantes sont bien parenthésées :

- (3 - (2 - 9))
- 1 + 4

Les expressions suivantes ne sont pas bien parenthésées :

- ((3) : une des deux parenthèses n'a pas été fermée.
- (4)) : on ferme une parenthèse qui n'a pas été ouverte.

Exercice 10:

- Écrire une fonction **length()** qui prend en paramètre une chaîne de caractère et qui retourne sa longueur.
- Écrire un programme qui lit deux chaînes de caractères et qui concatène la plus courte à la plus grande.

Exercice 11:

Écrire une fonction **Decal_lettre_chaine** qui décale de n positions les lettres d'une chaîne de caractère **ch** de longueur **L** (**on suppose que n < L**). Le résultat est rangé dans une chaîne CHD. L'entête est:

Fonction Decal_lettre_chaine(ch[]:caractere,L:Entier,n:Entier,CHD[]:caractere):vide

Exemple: si ch = "Rattrapage" et n=2 alors CHD = "geRattrapa"

Exercice 12 : Traitement des matrices à coefficients dans Z [Examen 2017]

- Écrire une fonction qui retourne la valeur 1 si une matrice M, de taille NxN, passée en paramètre est symétrique et retourne la valeur 0 sinon.
- Une matrice triangulaire supérieure** est une matrice carrée dont toutes les valeurs au-dessous de la diagonale principale sont nulles.

Ex :

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

Écrire 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

3. Une **matrice triangulaire inférieure** est une matrice carrée dont les valeurs au-dessus de la diagonale principale sont nulles.

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

Écrire 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

4. 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
5. Une **matrice diagonale** est une matrice carrée dont les coefficients en dehors de la diagonale principale sont nuls.

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

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

6. Écrire une fonction **PRODIAG** qui calcul le produit de deux matrices diagonales A et B et le met dans une matrice C. Les matrices A,B et C sont de taille **nxn**
 Fonction PRODIAG(A[][]:Entier, B[][]:Entier, C[][]:Entier, n:Entier) : Vide
7. Une matrice de Toeplitz est une matrice carrée dont les coefficients sur chaque diagonale descendant de gauche à droite sont les mêmes.

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

Écrire 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

8. Une matrice de **Hankel** est une matrice carrée dont les valeurs le long des diagonales ascendantes de gauche à droite sont constantes.

$$\text{Ex : } \begin{pmatrix} 3 & 7 & -1 & 2 & 6 \\ 7 & -1 & 2 & 6 & 18 \\ -1 & 2 & 6 & 18 & 4 \\ 2 & 6 & 18 & 4 & -5 \\ 6 & 18 & 4 & -5 & 125 \end{pmatrix}$$

Écrire une fonction **HANKEL()** qui retourne 1 si une matrice carrée M de taille nxn est de Hankel, sinon retourne 0. L'entête de la fonction est :

Fonction HANKEL(M[][] : Entier, n : Entier) : Entier

Exercice 13: [4 points] [~ 25 minutes]

- Écrire une fonction qui supprime les espaces, s'ils existent, au début d'une chaîne de caractères.
Exemple 1 : « SMI3 – 2018/2019 » deviendra «SMI3 – 2018/2019»
- Écrire une fonction qui supprime les espaces, s'ils existent, à la fin d'une chaîne de caractères.
- En déduire un programme qui lit une chaîne de caractère, puis supprime les espaces au début et à la fin de cette chaîne.

Exercice 14: [10 points][~ 60 minutes]

On considère qu'un texte est une chaîne de caractère suffisamment longue. Soient les suppositions suivantes:

- Un texte peut être vide.
- Si un texte n'est pas vide, il est composé d'une ou plusieurs phrases.
- Une phrase commence par une lettre ou un chiffre et se termine par un point. Il n'y a pas d'espace avant le point.
- Les phrases d'un texte sont séparées par un espace.
- Les mots d'une phrase sont séparés par un espace.

- Écrire une fonction **NPT** qui retourne le nombre de phrases d'un texte.
- Écrire une fonction **NMT** qui retourne le nombre de mots d'un texte.
- Écrire la fonction **Txt_Tab** qui met les phrases du texte **TX** dans **TCH[]**, un tableau de chaînes de caractères. Les chaînes de caractères, éléments du tableau **TCH[]**, n'ont pas nécessairement la même longueur. La fonction **Txt_Tab** doit avoir l'entête suivant:
Fonction **Txt_Tab**(TX : chaîne de caractères, TCH[] : chaînes de caractères) : vide

Exemple 3: si TX = "Aujourd'hui c'est l'examen d'Algorithmique. Bonne chance. Chacun pour soi et Dieu pour tous." alors il faut que le tableau TCH soit tel que :

TCH[0] = "Aujourd'hui c'est l'examen d'Algorithmique."

TCH[1] = "Bonne chance."

TCH[2] = "Chacun pour soi et Dieu pour tous."

Indication: On accepte l'affectation d'une chaîne de caractères à une autre, i.e $ch1 \leftarrow ch2$

- En déduire une fonction **Inv_Txt** qui inverse l'ordre des phrases d'un texte.
Exemple 4: le text TX dans l'exemple 3 deviendra TX = "Chacun pour soi et Dieu pour tous. Bonne chance. Aujourd'hui c'est l'examen d'Algorithmique."

Exercice 15: [3 points][~ 15 minutes] (Examen 2020)

Le triangle de Pascal est une matrice triangulaire inférieure telle que :

- contient 1 sur la diagonale,
- contient 1 sur le premier élément de chaque ligne
- les autres éléments sont construits comme indiqué sur l'exemple ci-dessous :

Exemple du TRIANGLE DE PASCAL de degré 5 :

N=0	1					
N=1	1	1				
N=2	1	2	1			
N=3	1	3	3	1		
N=4	1	4	6	4	1	
N=5	1	5	10	10	5	1

Écrire une fonction **TrPascal** qui prend en paramètre un entier N et une matrice triangulaire inférieure MP de dimension N+1 et retourne 1 si MP représente un TRIANGLE DE PASCAL de degré N et 0 sinon.