

Filière : TDI

Niveau : TS

Durée : 2h

Intitulé du module : Techniques de programmation structurée

Barème : / 40

**I) : Partie théorique (9/40 points)**

- 1) Citer quelques avantages des fonctions et procédures. (2 pts)
- 2) Quelle est la différence entre une structure et un tableau ? (2 pts)
- 3) On considère l'algorithme suivant :

**DEBUT**

**Variables N, I, Q, S: Entier**

**Lire(N)**

**$I \leftarrow N$**

**$S \leftarrow 0$**

**TantQue  $I \neq 0$  faire**

**$Q \leftarrow I \text{ Mod } 10$**

**$S \leftarrow S*10 + Q$**

**$I \leftarrow I / 10$**

**FinTanQue**

**Afficher (S)**

**FIN.**

- a) Dérouler cet algorithme pour **N = 52** puis pour **N = 123** et donner la valeur de **S** pour chacune des 2 valeurs de N. (4 pts)

**N = 52**

Q	S	I

Pour N = 52 → S=?

**N = 123**

Q	S	I

Pour N = 123 → S=?

- b) Que fait cet algorithme ? (1 pt)

## **II) : Partie pratique (31/40 points)**

### **Exercice 1 (6 pts)**

Ecrire un algorithme qui lit la taille N d'un tableau T du type entier, puis permet de remplir le tableau par des valeurs entrées au clavier et de supprimer les éléments pairs du tableau.

Afficher ensuite le tableau après suppression ainsi que le nombre d'éléments supprimés.

### **Exemple :**

Tableau initiale.

4	8	5	3	9	6	8	1	3
---	---	---	---	---	---	---	---	---

Tableau après suppression des données paires et compression des éléments restants.

5	3	9	1	3
---	---	---	---	---

### Exercice 2 (7 pts)

Ecrire un algorithme qui demande à l'utilisateur de taper des entiers positifs et qui calcule et affiche la moyenne de leur multiplication. Lorsqu'on tape une valeur négative, le programme affiche **RESSAYER** et demande de retaper une valeur. Lorsqu'on tape 0, cela signifie que le dernier entier a été tapé. On affiche alors la moyenne. Si le nombre d'entiers tapés est égal à 0, on affiche **ERREUR**.

**Exemple :**

8    9        10    567    2        6        11    0

On affiche

**$(8 * 9 * 10 * 567 * 2 * 6 * 11) / 7 = 7698240$**

### Exercice 3 (9 pts):

Les diviseurs propres d'un entier N sont tous les entiers autres que N qui le divisent.

Exemple : les diviseurs propres de 10 sont 1, 2 et 5

Un Triplet d'entiers (A, B, C) est dit **amiables** si et seulement si **la somme des diviseurs propres de l'un est égale à la somme des diviseurs propres des deux autres**.

Exemple : Le Triplet (5, 8, 10) est un Triplet d'entier amiables, en effet :

N	Diviseurs propres	Somme Diviseurs propres	<b>8=1+7</b>
5	1	1	
8	1, 2, 4	7	
10	1, 2, 5	8	

- 1) Ecrire une fonction qui reçoit comme paramètre trois entiers positifs (A,B,C) et retourne 1 si les trois entiers forment un Triplet d'entiers Amiables, dans le cas contraire la fonction doit retourner 0. **(7pts)**
- 2) Créer un programme principal qui demande 3 entiers A, B et C et qui vérifie si le Triplet (A,B,C) est amiable. Le programme doit utiliser la fonction précédente. **(2pts)**

**Exercice 4 (9pts) :**

Ecrire un algorithme qui permet de :

- 1) Lire une matrice de L lignes et C colonnes **(1,5 pt)**
- 2) Afficher ensuite cette matrice ligne par ligne. **(1,5 pt)**
- 3) Calculer et afficher la somme des éléments de la diagonale principale **(2 pts)**
- 4) Afficher un message qui précise si elle est triangulaire supérieure ou pas. Une matrice est triangulaire supérieure si toutes les valeurs au-dessous de la première diagonale sont égales à zéro. **(4 pts)**

Exemple de matrice triangulaire supérieure 4 x 4 :

<b>5</b>	3	-61	9
<i>0</i>	<b>2</b>	11	4
<i>0</i>	<i>0</i>	<b>15</b>	-7
<i>0</i>	<i>0</i>	<i>0</i>	<b>0</b>

Les éléments de la première diagonale sont en gras. Les éléments au-dessous de cette diagonale sont en italique. Dans cet exemple, ces derniers éléments sont tous égaux à zéro et donc, c'est une matrice triangulaire supérieure.