

**CORRIGÉ****PARTIE I (6 points)****Exercice 1 (4 points)**

- 1) 8
- 2) 17
- 3) La fonction retourne la somme des chiffres qui composent l'entier **n**.

**Exercice 2 (2 points)**

Proposition	Réponse
Un tableau de réels peut être rempli par des entiers.	<i>Vrai</i>
Le compteur d'une structure répétitive complète doit être de type scalaire.	<i>Vrai</i>
Les opérateurs <b>DIV</b> et <b>MOD</b> peuvent être appliqués sur les nombres réels.	<i>Faux</i>
<b>Efface (ch,longueur(ch)-1,2)</b> efface les deux derniers caractères de la chaîne <b>ch</b> .	<i>Vrai</i>

**PARTIE II (14 points)**

*Analyses : 8 points, Algorithme 4 points, paramètres et objets Locaux et Globaux : 2 points*

**ANALYSE DU PROGRAMME PRINCIPAL :**

- NOM = Valable
- 2. Résultat= PROC Affiche(n)
- 1. PROC Saisie(n)
- 3. Fin Valable

**T.D.O**

Objet	Type/Nature	Rôle
n	Entier	Le nombre d'éléments
Affiche	Procédure	Procédure de d'affichage
saisie	Procédure	Procédure de saisie

### **ALGORITHME DU PROGRAMME PRINCIPAL :**

- 0) Début Valable
- 1) PROC Saisie(n)
- 2) PROC Affiche(n)
- 3) Fin Valable

### **ANALYSE DE LA PROCEDURE SAISIE :**

- ```
DEF PROC Saisie (var m : entier)
2. Résultat= (m)
1. m= [] Répéter
    m = donnée
    Jusqu'à (m >= 1000) et (m <= 9999)
3. Fin Saisie.
```

### **ALGORITHME DE LA PROCEDURE SAISIE :**

- ```
0) DEF PROC Saisie (var m : entier)
1) Répéter
    Lire(m)
    Jusqu'à (m >= 1000) et (m <= 9999)
2) Fin Saisie
```

### **ANALYSE DE LA PROCEDURE AFFICHE :**

- ```
DEF PROC Affiche (p : entier)
2) Résultat = Affich
1) Affich = [] Si Valable (p) = Vrai Alors
    Ecrire (p, "est valable")
    Sinon
    Ecrire (p, "n'est pas valable")
    FinSi
3) Fin Affiche
```

### **ALGORITHME DE LA PROCEDURE AFFICHE**

- ```
0) DEF PROC Affiche (p : entier)
1) Si Valable(p) = vrai Alors
    Ecrire (p, "est valable")
    Sinon
    Ecrire (p, "n'est pas valable")
    FinSi
2) Fin Affiche
```

## ANALYSE DE LA FONCTION VALABLE :

DEF FN Valable (k : entier): booléen

3) Résultat = Valable

2) Valable = [] Si (c mod m = 0) et (d mod m = 0) et (u mod m = 0) Alors

    Valable ← vrai

    Sinon

    Valable ← faux

    FinSi

1) m ← k div 1000

    c ← k mod 1000 div 100

    d ← k mod 100 div 10

    u ← k mod 10

4) Fin Valable.

### T.D.O

Objet	Type/Nature	Rôle
m	entier	Chiffre des milliers.
c	entier	Chiffre des centaines.
d	entier	Chiffre des dizaines.
u	entier	Chiffre des unités.

## ALGORITHME LA FONCTION VALABLE :

0) DEF FN Valable (k: entier) : booléen

1) m ← k div 1000

    c ← k mod 1000 div 100

    d ← k mod 100 div 10

    u ← k mod 10

2) Si (c mod m = 0) et (d mod m = 0) et (u mod m = 0) Alors

    Valable ← vrai

    Sinon

    Valable ← faux

    FinSi

3) Fin Valable