



## TP 5 ASD1

### Exercice 1

```

Algorithme Exemple_5 ;
VAR n, som : entier;
Procedure Somme (var a:entire, var s:entier);
Debut
    s← 0 ;
    Tantque (a<>0) faire
        s← s + a;
        a← a-1;
    FinTantque;
Fin;
Debut
    Lire (n) ;
    Somme(n,som) ;
    Ecrire(som);
    n ← n+2;
    Somme(n,som);
    Ecrire(som) ;
Fin.
    
```

1. Ecrire cet algorithme en langage C.
2. Quelles sont les valeurs affichées à l'écran quand les valeurs de **n** lues sont ?

Valeur de <b>n</b>	2	3	4
Valeur de <b>som</b> après le premier appel à <b>Somme</b>			
Valeur de <b>som</b> après le deuxième appel à <b>Somme</b>			

3. Comment modifier le programme pour avoir des résultats corrects de **som**.

### Exercice 2

1. Ecrire une fonction **puiss** qui reçoit deux entiers **a, b** en paramètre et retourne l'élévation de **a** à une puissance **b**.
2. Ecrire une fonction **facto** qui reçoit un entier **a** en paramètre et retourne sa **factorielle**.
3. Ecrire, en utilisant **les deux fonctions précédentes**, le programme qui permet de calculer la formule ci-dessous :

$$\pi = \frac{2}{1!} + \frac{(1!)^2 2^2}{3!} + \frac{(2!)^2 2^3}{5!} + \frac{(3!)^2 2^4}{7!} + \dots = \sum_{i=0}^N \frac{2^{i+1}(i!)^2}{(2i+1)!}$$

### Exercice 3

Un entier positif **n** est appelé **parfait** si la somme de **ses diviseurs**, sauf **lui-même**, est égale à **n**.

**Exemple :** n=6, 1 + 2 + 3 = 6, donc **6 est parfait**. n=8, 1+2+4= 7≠8, donc 8 n'est pas **partait**

1. Écrire en langage C une fonction **parfait, booléenne**, qui retourne **vrai** si un entier **n** passé en paramètre est un nombre **parfait, faux** sinon.
2. Écrire le **programme principal** permettant d'afficher la liste des nombres **parfaits** compris entre **1 et 10000**. On utilisera le résultat renvoyé par la **fonction précédente**.

3. Transformer la fonction *parfait* en une **procédure** en signalant **toute modification** au niveau du **programme principal**.

#### **Exercice 4**

1. Ecrire une **fonction récursive** permettant de calculer la somme :  $1+2+3+\dots+n$ , tel que **n** un nombre entier positif passé en paramètre.
  2. Ecrire une **fonction récursive** permettant de calculer le factoriel d'un nombre **n** passé en paramètre.
  3. Ecrire une **fonction récursive** permettant de calculer la puissance  $a^b$ , tel que **a, b** deux nombres entiers positifs passés en paramètres.
  4. Ecrire une **fonction récursive** permettant de calculer la **somme de chiffres d'un nombre a** entier positif passé en paramètre.
- ❖ Ecrire le programme principal qui appelle chaque fonction.