TD 5 ASD1 : Les sous-programmes

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. Quelles sont les valeurs affichées à l'écran quand la valeur de n lue est 3 ?
2. Comment modifier l'algorithme pour que les valeurs affichées à l'écran soient successivement 6 et 15.

Exercice 2

Donner le résultat des quatre algorithmes ci-dessous.

```
Algorithme exemple_1 ;
var x :entier ;
Procedure Modifie() ;
debut
  x←1 ;
fin ;
Debut
  x←0 ;
  Modifie() ;
  Ecrire(x) ;
Fin.
```

```
Algorithme exemple_2 ;
var x :entier ;
Procedure Modifie () ;
Var x :entier
Debut
  x←1 ;
fin ;
Debut
  x←0 ;
  Modifie() ;
  Ecrire(x) ;
Fin.
```

```
Algorithme exemple_3 ;
var x :entier ;
Procedure Modifie (y :entier) ;
debut
  y←1 ;
fin ;
Debut
  x←0 ;
  Modifie(x) ;
  Ecrire(x) ;
Fin.
```

```
Algorithme exemple_4 ;
var x :entier ;
Procedure Modifie (Var y :entier) ;
debut
  y←1 ;
fin ;
Debut
  x←0 ;
  Modifie(x) ;
  Ecrire(x) ;
Fin.
```

Exercice 3

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,** l'algorithme principal qui permet de calculer la formule ci-dessous :

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

Exercice 4

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

Exemple : n=6, la somme de diviseurs de n sauf **lui-même** = 1 + 2 + 3 = 6**,** donc **6 est *k-parfait.***

1. Ecrire en langage algorithmique une fonction ***parfait,* booléenne**, qui retourne ***vrai*** si un entier ***n*** passé en paramètre est un nombre ***parfait***, ***faux*** sinon.
2. Ecrire l'algorithmique principal permettant d'afficher la liste des nombres ***parfaits*** compris entre **1 et 20000.** On utilisera le résultat renvoyé par **la fonction précédente.**
3. Transformer la fonction ***parfait*** en une **procédure** en signalant **toute modification** au niveau du **l'algorithme principal.**

Exercice 5

Ecrire une **fonction récursive** permettant de calculer :

1. La somme **1+2+3+ … + n**, tel que **n** un nombre entier positif passé en paramètre.
2. la **somme de chiffres d'un nombre** a entier positif passé en paramètre.
3. Le n$^{\text{ième}}$ terme de la suite de Fibonacci définit comme suit:

$$\begin{cases} U_0=0, \\ U_1=1, \\ U_n=U_{n-1}+U_{n-2},\ \forall\ n \geq 2. \end{cases}$$