

**Corrigé-Devoir De Synthèse n°2****Algorithmique & Programmation****Exercice n°1 : (3.75 points) :**

Les algorithmes correspondants aux fonctions F1, F2, F3 et F4 permettent de :

- F1 et F2 , convertir un nombre décimal vers son équivalent en base B
- F3, convertir un nombre d'une base B quelconque vers son équivalent décimal
- F4, convertir un nombre d'une base B inférieure à 10 vers son équivalent décimal

**Réponses aux questions :** Mettre V si la proposition est correcte et F sinon

N° question :	Réponse			
1	F	V	F	
2	V	F	F	
3	V	V	F	F
4	F	F	F	V
5	F	F	V	F

**Exercice n°2 : (2.5 points)**

**Algorithme d'un module permettant de vérifier la divisibilité d'un nombre N par 19.**

Fonction Divis\_19(N : Entier) : Booléen

Début

Répéter

$N \leftarrow (N \text{ div } 10) + 2 * (N \text{ mod } 10)$

Jusqu'à ( N < 38)

Retourner ((N=0) OU (N=19))

Fin

T.D.O

O	T/N
-	-

**Exercice n°3 : (2.5 points)**

Soit la formule suivante :

$$e = 1 + \frac{1}{1} + \frac{1}{1 \times 2} + \frac{1}{1 \times 2 \times 3} + \frac{1}{1 \times 2 \times 3 \times 4} + \frac{1}{1 \times 2 \times 3 \times 4 \times 5} + \dots = \sum_{n=0}^{\infty} \frac{1}{n!}$$

**Ecrire un algorithme d'un module permettant de calculer une valeur approchée de e à 10<sup>-5</sup> près.**

Fonction e\_app (epsilon : réel) : réel

Début

$S1 \leftarrow 1, i \leftarrow 1$

Répéter

$S2 \leftarrow S1$

$S1 \leftarrow S2 + 1/\text{Fact}(i)$

$i \leftarrow i + 1$

Jusqu'à ( ABS(S1 -S2) < epsilon)

Retourner S1

Fin

T.D.O :

O	T/N
S1, S2	Réel
i	Entier
Fact	Fonction

Fonction fact( n : entier) : entier

Début

$F \leftarrow 1$

Pour i de 2 à n faire

$F \leftarrow F * i$

Fin

Retourner F

Fin

### Exercice n°4 : (3.75 points)

Exemple : pour  $n = 4$

	0	1	2	3	
0	60	48	16	34	← la 1ère ligne remplit aléatoirement
1	12	16	2		← PGCD(16, 34)
2	4	2			
3	2				← PGCD (60, 48, 16, 34)

**Algorithme d'un module permettant de calculer et afficher le PGCD de N entier positifs.**

Procédure Remplir (n : entier , @ M : Mat)

Début

Pour j de 0 à n-1 faire

$M[0,j] \leftarrow \text{Aléa}(10,100)$

Pour i de 1 à n-j-1 faire

$M[i,j] \leftarrow \text{PGCD}(M[i-1,j], M[i-1,j+1])$

Fin pour

Fin pour

Ecrire("Le PGCD est : ",  $M[n-1,0]$ )

Fin

⇒ Proposer un algorithme solution de la fonction PGCD (méthode de différence ou de division euclidienne)

**T.D.O :**

O	T/N
i,j PGCD	Entier Fonction

### Exercice n°5 : (7.5 points)

#### 1) Algorithme du programme principal

Algorithme calculette

Début

Saisir( n )

Remplir (n,T, M)

Afficher (n,M)

Fin

TDNT :

Type
TAB = tableau de 10 chaines
MAT=Tableau de n*3 chaines

TDO :

O	T/N
N	Entier
T	TAB
M	MAT
Saisir	Procédure
Remplir	Procédure
Afficher	Procédure

#### 2) Algorithmes des sous-programmes :

Procédure Saisir( @ n : entier )

Début

Répéter

Ecrire ("donner le nombre de lignes : ")

Lire (n)

Jusqu'à (n > 2)

Fin

O	T/N
-	-

Procédure afficher (n : entier, M : mat )

Début

Pour i de 0 à n-1 faire

Pour j de 0 à 2 faire

Ecrire (M[i,j], " | ")

Fin pour

Fin pour

Fin

O	T/N
i,j	Entier

Procédure remplir (n : entier, T : TAB, @ M : MAT)

Début

Pour i de 0 à n-1 faire

$M[i,2] \leftarrow \text{Addition}(M[i,0], M[i,1], T)$

Fin pour

Fin

O	T/N
i Addition	Entier Fonction

Fonction Addition (ch1,ch2 : chaine, T : TAB) : chaine

Début

X ← équivalent(ch1,T)

Y ← équivalent( ch2,T)

Ch ← somme(x, y, T)

Retourner ch

Fin

Fonction équivalent (ch: chaine, T : TAB) : entier

Début

X ← 0

Tantque (ch <> "") faire

Ch1 ← sous-chaine (ch, 0, 7)

i ← Recherche( ch1, T)

X ← X\*10 + i

Effacer (ch, 0, 7)

Fin tq

Retourner X

Fin

O	T/N
X, Y	Entier
Ch	chaine
Equivalent	Fonction
Somme	Fonction

O	T/N
X, i	Entier
Ch1	chaine
Recherche	Fonction

Fonction Somme (x, y : Entier , T : TAB) : chaine

Début

A ← x + y

Res ← ""

Tantque (A <> ) faire

Res ← T[A mod ] + res

A ← A div

Fin

Retourner Res

Fin

O	T/N
A	Entier
Res	chaine

Fonction Recherche (ch : chaine , T : TAB) : Entier

Début

i ← -1

Tant que (T[i+1] <> ch) faire

i ← i + 1

fin tq

retourner i

Fin

O	T/N
i	Entier