

Année universitaire : 2022 - 2023

Algorithme et pseudo-code

Examen

Session 2

Durée : 1h 30

Aucun document n'est autorisé.

**Exercice 1 : (6 points)**

Soit le sous algorithme ou sous-programme suivant :

La suite de Fibonacci est définie par  $(F_n)_{n \in \mathbb{N}}$  :

$$\begin{cases} F_0 = 1 \\ F_1 = 1 \\ F_{n+2} = F_{n+1} + F_n, \forall n \in \mathbb{N} \end{cases}$$

Entrées :  $n$  entier, indice du terme de la suite de Fibonacci

Résultat : le terme de rang  $n$  de la suite de Fibonacci

```
1 fonction Fibb(n)
2 début
3   si  $n=0$  alors
4     retourner 1
5   sinon si  $n=1$  alors
6     retourner 1
7   sinon
8     retourner  $\text{Fibb}(n-1) + \text{Fibb}(n-2)$ 
9 fin
```

- 1- Utiliser cette fonction dans un algorithme
- 2- Traduire en langage C, l'algorithme.
- 3- Donner la trace de  $\text{Fibb}(5)$

**Exercice 2 : (8 points)**

- 1- Ecrire un algorithme qui demande à l'utilisateur deux chiffres, et affiche à l'écran le plus grand en valeur absolue. Si les deux nombres ont la même valeur absolue mais pas le même signe, le programme affichera celui qui est positif
- 2- Ecrire un algorithme qui vérifie l'admissibilité d'un étudiant qui passe trois examens. Il est déclaré admis si :
  - Soit, il a au moins 9 points à chaque examen.
  - Soit, la moyenne des trois examens est au moins égale à 10 points et la plus basse



note est au moins égale 8 points.

S'il n'est pas admis alors il est refusé

- 3- Ecrire un algorithme qui demande à l'utilisateur de saisir deux entiers positifs (le programme devra les stocker dans deux variables nommées *dividende* et *diviseur*) ; puis qui calcule le quotient et le reste de la division euclidienne de *dividende* par *diviseur*.

Attention, le programme devra vérifier que le dividende est positif ou nul, et que le diviseur est strictement plus grand que 0, et devra afficher un message d'erreur le cas échéant.

### Exercice 3 : (6 points)

1. Soit la suite de Tribonacci

$$u_0 = u_1 = 0, u_2 = 1 \text{ et}$$

$$u_{n+3} = u_{n+2} + u_{n+1} + u_n$$

- a. Ecrire la version récursive et itérative de cette suite
  - b. Traduire la version itérative en langage C
2. Ecrire un programme qui, à partir d'une liste d'entiers, détermine le nombre maximum de la liste, et le nombre d'occurrences de cet élément.

**NB : Attention, la liste d'entiers ne devra être parcourue qu'une seule fois**