Travaux dirigés d'informatique 4

Serie $N°1$

NB :

1. ***Ces exercices sont extraits du support de cours "Algo de Base" du Dr NDONG NGUEMA***.
2. ***Les algorithmes demandés seront donnés sous forme de fonctions ou procédures algorithmiques.***
3. ***Il est vivement conseillé d'implementer et tester ces algorithmes sous :*** C**,** $Matlab$ ***ou*** $Python$**.**
4. ***On écrira une procédure permettant de lire les termes d'une suite contenus dans un fichier et les stocke dans une structure de donnée de type :*** **Suite**.

Exercice 1 :

$$\forall\, n \in \mathbb{N},\ u_{n+1} = n \cdot sin(u_n) + 2\,. \tag{1}$$

1. *Concevoir et écrire une fonction algorithmique* **non récursive** *qui calcule le terme de rang n arbitraire d'une suite* (u_n) *satisfaisant la récurrence* (1).
2. *Concevoir et écrire une fonction algorithmique* **récursive** *qui calcule le terme de rang n arbitraire d'une suite* (u_n) *satisfaisant la récurrence* (1).
3. *Concevoir et écrire une fonction ou une procédure algorithmique qui calcule tous les termes jusqu'à un rang n arbitraire d'une suite* (u_n) *satisfaisant la récurrence* (1).

Exercice 2 :

$$\forall\, n \in \mathbb{N}\ (n \geq 5),\ w_n = \frac{sin(w_{n-1}) + sin(w_{n-2}) + 5}{\sqrt{n} - 2}\,. \tag{2}$$

1. *Concevoir et écrire une fonction algorithmique* **non récursive** *qui calcule le terme de rang n arbitraire d'une suite* (w_n) *satisfaisant la récurrence* (2).
2. *Concevoir et écrire une fonction algorithmique* **récursive** *qui calcule le terme de rang n arbitraire d'une suite* (w_n) *satisfaisant la récurrence* (2).
3. *Concevoir et écrire une fonction ou une procédure algorithmique qui calcule tous les termes jusqu'à un rang n arbitraire d'une suite* (w_n) *satisfaisant la récurrence* (2).

Exercice 3 : **(Recherche du maximum d'une suite finie)**

Écrire un programme en L.E.A. qui détermine le maximum d'une suite finie de nombres réels lus dans un fichier sur le disque.

N.B. *Pour la recherche du maximum proprement dite, proposer une* ***approche non récursive*** *et une* ***approche récursive****.*

Exercice 4 : **(Recherche du maximum et du minimum d'une suite finie)**

Écrire un programme en L.E.A. qui détermine le maximum et le minimum d'une suite finie de nombres réels lus dans un fichier sur le disque.

N.B. *Pour la recherche du maximum et du minimum proprement dite, proposer une* ***approche non récursive*** *et une* ***approche récursive****.*

Exercice 5 : **(Recherche des 2 plus petits éléments d'une suite finie)**

Écrire un programme en L.E.A. qui détermine les 2 plus petits éléments (au sens de la relation $\leqslant$*) d'une suite finie de nombres réels lus dans un fichier sur le disque.*

N.B. *Proposer une* ***approche non récursive*** *et une* ***approche récursive****.*

Exercice 6 : **(Recherche (simultanée) des 3 plus petits éléments d'une suite finie)**

*Idem que l'****Exercice 5****, mais pour les 3 plus petits éléments d'une suite finie de réels.*

Exercice 7 : **(Recherche du maximum d'une section d'une suite finie)**

Une ***section*** *d'une suite finie est toute sous-suite d'éléments consécutifs de cette suite.*

Écrire un programme en L.E.A. trouvant le maximum de la section d'une suite finie de nombres réels lus dans un fichier sur le disque, section de la suite d'un indice IDEB donné à un indice IFIN donné.

N.B. *Proposer une* ***approche non récursive*** *et une* ***approche récursive****.*

Exercice 8 : **(Sections monotones d'une suite finie)**

1. *Une* ***section strictement croissante*** *d'une suite finie est toute sous-suite strictement croissante d'éléments consécutifs de cette suite.*
2. *On définit de manière analogue une* ***section strictement décroissante*** *d'une suite finie.*
3. *Un* ***palier*** *d'une suite finie est tout sous-suite constante d'éléments consécutifs de cette suite.*

4. *La **longueur d'une section** est son nombre d'éléments.*

Écrire un programme en L.E.A. qui découpe et affiche une suite suivant ses sections strictement croissantes, ses sections strictement décroissantes, et ses paliers, dans leur ordre d'apparition dans la suite.

NOTA : *Pour l'affichage, une section par ligne, et précédée par un marqueur indiquant s'il s'agit d'un palier, d'une section strictement croissante ou strictement décroissante.*
Effectuer ensuite un affichage du même type, mais par ordre décroissant des longueurs des sections

Exercice 9 : (Sections unimodales ou collines d'une suite finie)

1. *Une **section unimodale croissante** ou **colline** d'une suite finie est toute sous-suite d'éléments consécutifs de cette suite constituée d'une section strictement croissante suivie d'une autre strictement décroissante.*
2. *Dans cette définition, il est entendu ici que l'ensemble des indices d'une section unimodale ne peut pas être contenu dans l'ensemble des indices d'une autre section unimodale.*
3. *Le **sommet** de la colline est l'élément le plus grand de la section unimodale.*

Écrire un programme en L.E.A. qui découpe et affiche toutes les sections unimodales (s'il y en a, et en précisant leurs sommets), dans leur ordre d'apparition (puis dans l'ordre croissant de leurs longueurs), d'une suite finie donnée.

N.B. *Proposer une **approche non récursive** et **une approche récursive.***

Exercice 10 : (Fusion ordonnée de 2 suites finies strictement croissantes)

Écrire un programme en L.E.A. qui prend, en entrée deux suites finies strictement croissantes lues, chacune, dans un fichier ;
et renvoie, comme résultat, la suite finie strictement croissante obtenue en fusionnant les 2 précédentes de manière à maintenir l'ordre strictement croissant. Pour cela, on examinera 2 cas :

1. ***Les 2 suites initiales n'ont aucun élément en commun.***
2. ***Les 2 suites initiales pourraient avoir un ou des élément(s) en commun.***

Exercice 11 : (Insertion ordonnée d'un élément dans une suite finie ordonnée)

Écrire un programme en L.E.A. qui insère un nouvel élément (s'il n'y est pas encore) dans une suite finie strictement croissante, de telle sorte que le résultat reste une suite finie strictement croissante.