

## LIF 1 – Lundi 18 avril 2011

Durée 1h

*Les documents, calculatrice, téléphone portable sont interdits**La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.**Vous veillerez à respecter les notations et les règles d'écriture des algorithmes vues en cours et en TD.**Soyez attentifs aux indices des tableaux et prenez garde à ne pas dépasser la fin des tableaux.**Le barème est donné à titre indicatif.***Partie A : Algorithmique (10 points)**Exercice I

Soit la suite de Syracuse, définie comme suit :

$$u_{n+1} = 1 + 3u_n \text{ si } u_n \text{ impair ;}$$

$$u_{n+1} = u_n / 2 \text{ si } u_n \text{ pair.}$$

Quel que soit le nombre entier  $u_0$  strictement positif, cette suite finit par engendrer le nombre 1.

- 1- Écrire en langage algorithmique un sous-programme permettant de saisir une valeur comprise entre 1 et 20 (dimension maximale du tableau déclaré) ; on s'assurera que la valeur retournée appartient bien à l'intervalle voulu.
- 2- Écrire en langage algorithmique une fonction `syracuse` qui remplit un tableau avec les termes de la suite de `syracuse` précédemment définie à partir d'une valeur `lambda`, donnée par l'utilisateur, et jusqu'à l'obtention de la valeur 1. Cette fonction devra retourner le nombre d'itérations au bout duquel la suite engendre la valeur 1.

Exemple : Pour `lambda = 11` : Il faudra 14 itérations pour engendrer 1 et le tableau final sera :

11	34	17	52	26	13	40	20	10	5	16	8	4	2	1				
----	----	----	----	----	----	----	----	----	---	----	---	---	---	---	--	--	--	--

- 3- Écrire en langage algorithmique une procédure `affiche_syracuse` permettant l'affichage du tableau obtenu ; attention, on n'affichera que les valeurs significatives du tableau (i.e. celles pour lesquelles le tableau a été rempli).
- 4- Écrire en langage algorithmique le programme principal permettant de saisir une valeur `lambda`, d'appeler la fonction `syracuse` et d'afficher le nombre d'itérations nécessaires ainsi que le tableau obtenu.

Exercice II

Écrivez en langage algorithmique une fonction booléenne qui teste si un tableau 1D passé en paramètre est trié ou non. La taille du tableau est également passée en paramètre.

T1 = 

1	2	5	9	12
---	---	---	---	----

`est_trie(T1, 5) → Vrai`T2 = 

5	8	6	2	9
---	---	---	---	---

`est_trie(T2, 5) → Faux`

## Partie B : Langage C (10 points)

### Exercice I

Définition : Étant donnés deux nombres réels  $a$  et  $b$ , le procédé qui à tout nombre  $x$  fait correspondre le nombre  $ax + b$  s'appelle une **fonction affine**. Cette fonction permet de définir dans le plan une **droite affine**.

Détermination des coefficients  $a$  et  $b$  : Si  $M(x_1, y_1)$  et  $N(x_2, y_2)$  sont deux points appartenant à la droite d'équation  $y = ax + b$ , alors on a :

$$a = \frac{y_2 - y_1}{x_2 - x_1}$$
$$b = y_1 - ax_1 = y_2 - ax_2$$

- 1- Écrire en langage C / C++ une **procédure** permettant, à partir des coordonnées de 2 points, de calculer et de retourner les coefficients  $a$  et  $b$  de la droite affine.
- 2- Donnez une traduction possible de l'entête de la procédure précédente en **fonction**
- 3- Écrire en langage C / C++ une **fonction booléenne** que retournera vrai si un point passé en paramètre appartient à une droite affine donnée et faux sinon.

### Exercice II

Écrivez en langage C une procédure qui remplit les  $N$  premières diagonales d'un tableau carré de  $N \times N$  (le tableau et  $N$  sont passés en paramètre) suivant l'anti-diagonale, avec les entiers de 1 à  $N(N+1)/2$ . Exemple, pour  $N=4$ , le résultat sera :

10			
6	9		
3	5	8	
1	2	4	7