TD informatique du chapitre 2 : Méthodes de programmation

Exercice 1 : On considère la suite définie par :

$$u_0 = 2 \text{ et } \forall n \in \mathbb{N},\ u_{n+1} = \sqrt{u_n^2 + 1}.$$

1. Ecrire une fonction qui prend comme argument un entier naturel n et qui affiche successivement : "pour $k = \ldots$, on a : $u_k = \ldots$" pour $0 \leq k \leq n$.
2. Ecrire une fonction qui prend comme argument un flottant m et qui renvoie le plus petit n tel que $u_n \geq m$.

Exercice 2 : Fonctions classiques

1. Ecrire une fonction qui prend comme argument un flottant et qui renvoie sa valeur absolue **sans** utiliser la fonction `abs` ou d'autres fonctions déjà programmées.
2. Ecrire une fonction qui prend comme argument un entier naturel n et qui renvoie $n!$.
3. Ecrire une fonction qui prend comme argument un réel positif x et qui renvoie sa partie entière, c'est-à-dire le plus grand entier inférieur ou égal à x.

Exercice 3 : Suite de Syracuse

On appelle suite de Syracuse la suite définie par :

$$u_0 \in \mathbb{N}^* \qquad \text{et :} \qquad \forall n \in \mathbb{N},\ u_{n+1} = \begin{cases} \dfrac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

La conjecture de Syracuse affirme que pour toute valeur de $u_0 \in \mathbb{N}^*$, il existe $N \in \mathbb{N}$ tel que $u_N = 1$.
Après que le nombre 1 a été atteint, la suite des valeurs 1, 4, 2, 1, 4, 2, ... se répète indéfiniment en un cycle de longueur 3, appelé cycle trivial. Cette conjecture, malgré la simplicité de son énoncé, n'a toujours pas été prouvée.
On définit, en supposant l'existence de tels entiers :

- **le temps de vol** de la suite comme étant le plus petit entier naturel n tel que $u_n = 1$
- **l'altitude maximale** de la suite comme étant la valeur maximale de la suite avant d'atteindre la valeur 1.

Par exemple à partir de $u_0 = 15$ on obtient :

u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}
15	46	23	70	35	106	53	160	80	40	20	10	5

u_{13}	u_{14}	u_{15}	u_{16}	u_{17}	u_{18}	$\cdots$
16	8	4	2	1	4	$\cdots$

Le temps de vol de cette suite de Syracuse est de 17 et son altitude maximale est de 160.

1. Ecrire une fonction `Syracuse(u0,n)`, qui prend en entrée deux entiers strictement positifs `u0` et `n`, et renvoie le terme u_n de la suite de Syracuse définie avec la valeur initiale $u0$.
2. Ecrire une fonction `TempsVol(u0)` qui prend en entrée un entier strictement positif `u0` et renvoie le temps de vol de la suite de Syracuse définie avec la valeur initiale `u0`.
   Remarque : on supposera vraie la conjecture de Syracuse évoquée plus haut. Ainsi, le temps de vol de la suite est bien défini.
3. Ecrire une fonction `altitude(u0)`, qui prend en entrée un entier strictement positif `u0` et renvoie l'altitude maximale de la suite de Syracuse définie avec la valeur initiale `u0`.
   Remarque : on supposera vraie la conjecture de Syracuse évoquée plus haut. Ainsi, l'altitude maximale de la suite est bien définie.

Exercice 4 : Jeu du nombre secret

Programmer le jeu du nombre secret selon le principe suivant : l'ordinateur choisi aléatoirement un entier entre 0 et 99. L'utilisateur doit deviner ce nombre en proposant un nombre. L'ordinateur doit donner, selon les cas une des réponses suivantes :

- Quelle est ta proposition ?
- Trop petit, recommence !
- Trop grand, recommence !
- Bravo, le nombre secret était …, tu as gagné en …essais.
- Perdu, tu as fait 101 essais !

On pourra utiliser, après avoir importé le module `random`, la commande `random.randint(a,b)` qui renvoie un entier aléatoire de $[\![a, b]\!]$.