	Classe préparatoire 1^{ère} année Projet informatique	
	Florent Devin	
	<i>Matière : Informatique</i>	<i>Date : Janvier 2017</i>
	Documents de cours et TD autorisés	<i>Durée de l'épreuve :</i> <i>Nombre de pages du sujet : 5</i>

1 Contexte général

Le « *L-system* » fut introduit par le biologiste *A. Lindenmayer* dans le but de modéliser la croissance des plantes, le principe de base étant qu'une graine contient déjà en elle la forme finale de la plante. Dans un *L-system*, la graine est composée de :

- un « *axiome* » : la forme première de la plante ;
- des « *règles de croissance* » : la façon de se développer.

L'axiome et les règles de croissance sont représentés par des suites de symboles. À la première étape (niveau 0) est l'axiome puis, à l'étape suivante, chaque symbole de l'axiome correspondant à une règle est remplacé par la suite des symboles représentant cette règle et ainsi de suite. De cette manière se construit une suite de symboles qui peut être interprétée par le tracé d'une figure à l'écran. Comme correspondance entre le symbole et l'image, nous utilisons un module spécifique de python (*turtle*¹) qui permet de déplacer une « *tortue* » sur l'écran en la faisant ou non tracer son chemin.

Ceux-ci sont définis par :

- un axiome représenté par une chaîne de caractères ;
- une (ou plusieurs) règle(s) de croissance représentée(s) par une chaîne de caractères :
 - la tête (symbole à remplacer) est un caractère ;
 - le corps (les caractères remplaçant la tête) est une chaîne de caractères.
- un angle exprimé en degrés.

Il faut aussi définir les caractères et leur équivalent en langage *turtle*. Par exemple :

1. `a : pd(); fd(taille);`
2. `b : pu(); fd(taille);`
3. `+: right(angle);`
4. `-: left(angle);`
5. `* : right(180);`
6. `[: mémoriser la position actuelle (à coder);`
7. `] : retourner à la dernière position mémorisée (à coder).`

Ainsi, dans le cas du « *flocon de Koch* », nous avons au niveau 0 :

- `axiome = "-a++a++a";`
- `regle = "a=a-a++a-a";`
- `angle = 60.`

Dans cet exemple, l'application de la règle de croissance à l'axiome conduira au *L-system* de niveau 1 suivant :

- `axiome = "-a-a++a-a++a-a++a-a++a-a";`
- `regle = "a=a-a++a-a";`
- `angle = 60.`

La figure 1 présente les trois premières étapes du développement du flocon de Koch.

Pour utiliser la librairie *turtle*, vous devez ajouter la ligne `from turtle import *` dans votre programme. Voici un exemple (tiré du site) :

1. <https://docs.python.org/3/library/turtle.html>

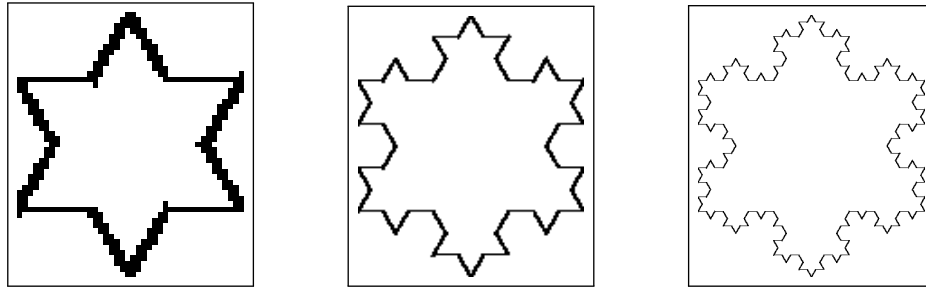


Figure 1 – Exemple de développements successifs

Code

```

from turtle import *
color('red', 'yellow')
while True:
    forward(200)
    left(170)
    if abs(pos()) < 1:
        break
done()

```

Remarques :

- la figure 2 présente la figure obtenue pour un développement au niveau 4 du L-system :
 - axiome = "--a";
 - regle = "a=aa+[+a[+a]-a+a]-[-a-[a]+a]";
 - angle = 30.
- la figure 3 présente la figure obtenue pour un développement au niveau 4 du L-system :
 - axiome = "---a";
 - regle = "a=aa+[+a-a-a]-[-a+a+a]";
 - angle = 22.50.
- la figure 4 présente la figure obtenue pour un développement au niveau 3 du L-system :
 - axiome = "a-a-a-a";
 - regle = "a=a+a*a+a+aa*aa+a+a*a+a";
 - angle = 90.

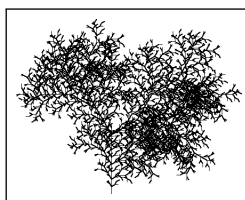


Figure 2 – Exemple de L-system

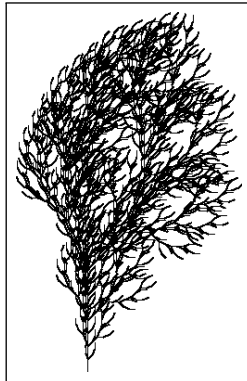


Figure 3 – Exemple de L-system

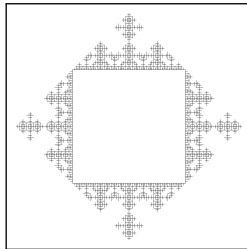


Figure 4 – Exemple de L-system

Les fichiers d'entrée auront la *forme* suivante :

Code

```

axiome = "---a"
regles =
    "a=aa+[+a[+a]-a+a]-[-a-[a]+a]"
    "b=bb+b-a+b-a-bb"
angle  = 60
taille = 10
niveau = 2
  
```

Remarques :

- Il ne peut y avoir qu'un seul axiome par fichier
- Un angle et un niveau devront être systématiquement présent.
- Il peut y avoir une ou plusieurs règles de transformation.
- L'ordre des définitions n'a pas d'importance.
- Toutes les règles de croissance s'appliquent d'un niveau à l'autre.

Exercice 1 (Travail).


Vous devez réaliser un programme qui

- lit un fichier d'entrée comportant la définition complet d'un « *L-system* » ;
- génère le « *L-system* » au niveau spécifié ;
- génère un programme `python` comportant l'ensemble des instructions permettant de tracer le « *L-system* ».


Ci-dessous, vous trouverez un ensemble de consignes qu'il vous faut impérativement respecter. Le non-respect de ces consignes entraînera systématiquement des pénalités. Voici la liste des consignes :

- Pour tout fichier d'entrée incorrect, votre programme doit s'arrêter et indiquer la raison de l'arrêt.
- Le fichier d'entrée sera demander explicitement dès le début de votre programme.
- Vous devrez afficher *UNIQUEMENT* l'ensemble des commandes permettant de générer la figure demandée sur le terminal. Tout affichage supplémentaire sera sanctionné.
- Vous pouvez éventuellement lancer le traçage du « *L-system* » directement via votre programme. Ceci ne vous dispense pas de générer la sortie.
- Votre programme ne doit pas faire d'autres affichages que ceux demandés.
- Vous n'oublierez pas de fournir les tests unitaires de votre programme.
- Les fonctions que vous coderez ne devront en aucun cas dépasser 30 lignes.

Par exemple, si votre fichier d'entrée est :

```
 Code  
axiome = "a"  
regles =  
    "a=-a++a++a"  
    "b=bb+b-a+b-a-bb"  
angle = 60  
taille = 50  
niveau = 1
```

L'affichage devra être (qui correspond donc à `-a++a++a`) :

```
 Code  
from turtle import *  
color('black')  
left(60);  
pd();fd(50);  
right(60);  
right(60);  
pd();fd(50);  
right(60);  
right(60);  
pd();fd(50);  
exitonclick();
```

Exercice 2 (Bonus).

Vous pouvez améliorer votre note, si vous traitez une des parties suivantes :

- a. Traitement de la ligne de commande :
 - Le fichier d'entrée doit pouvoir être spécifié via la ligne de commande (via l'option -i).Si le commutateur -i n'est pas présent, vous devez arrêter votre programme et indiquer à l'utilisateur la raison de l'arrêt. Dans ce cas, le fichier ne sera pas demandé à l'utilisateur
 - Un fichier de sortie doit pouvoir être spécifié via la ligne de commande (via l'option -o).
- b. Traitement d'autres types de fractale :
 - *Stochastic L-systems*
 - *Context sensitive L-systems*
 - *Parametric L-systems*
- c. Invention d'autres symboles permettant de modifier la tortue (épaisseurs du trait, couleur, ...)