Exercices
Semaine 9

Cours Turing

Note préliminaire : Pour ces exercices, vous aurez besoin du module matplotlib.pyplot pour représenter des graphes :

- insérez la commande suivante au début de vos fichiers :

```
from matplotlib.pyplot import *
```

- pour tracer le graphe d'une liste de nombres L, utilisez :

```
plot(L,".")
show()
```

- pour tracer l'histogramme des nombres composant cette même liste L, utilisez :

```
hist(L,nb_bins)
show()
```

où nb_bins est le nombre de "bins" (littéralement, de "paniers") de votre histogramme.

1 Méthode des carrés tronqués

Ecrivez un programme qui demande en entrée un nombre x à 8 chiffres (par exemple), et génère à partir de là une suite de nombres selon la méthode décrite en cours : pour calculer le prochain nombre de la liste, on calcule x^2 et on ne retient que les 8 chiffres du "millieu" de celui-ci. Il y a plusieurs façons de faire cela : nous vous laissons réfléchir...

Représentez l'histogramme de la liste des nombres ainsi générés. Qu'observez-vous ? (attention : l'observation dépend bien sûr du nombre x choisi au départ, ainsi que de la quantité de nombres générés)

2 Générateur à congruence linéaire

Ecrivez un programme qui demande en entrée trois paramètres m, a et b, puis qui génère une liste de nombres selon le schéma suivant :

$$x \to (ax + b) \,(\text{mod } m)$$

partant par exemple du nombre $x = 1$ au départ.

Représentez ensuite graphiquement :

- la liste des nombres ainsi générés

- leur histogramme

- le graphe avec en abscisse les points x de la liste L et en ordonnée les points correspondants $(ax + b) \,(\text{mod } m)$.

Note : Pour tracer un graphe de points dont les abscisses sont enregistrées dans une liste L et les ordonnées sont enregistrées dans une liste M, utilisez :

```
plot(L,M,".")
show()
```

Suivant les différents paramètres m, a et b choisis, qu'observez-vous ?

3 Algorithme Xorshift

Implémentez la version de l'algorithme Xorshift vue au cours pour générer des séquences de nombres aléatoires d'une taille d'au plus 32 bits chacun.

Attention : Pour garantir que chaque nombre produit ne dépasse pas la taille de 32 bits, il vous faudra ruser un peu (indication : utiliser l'opération ET peut s'avérer utilie).

Qu'observez-vous si au lieu d'effectuer pour chaque nouveau nombre les trois étapes de l'algorithme, vous n'en effectuez que deux (ou même qu'une seule) ?