Master Sciences, Technologies, Santé
Mention Mathématiques, spécialité Enseignement des mathématiques

Algorithmique et graphes, thèmes du second degré

Séance de travaux pratiques n° 2

Cette séance a pour but de se familiariser avec la programmation Python pour ce qui concerne l'algorithmique de base, la manipulation de listes et les aspects graphiques.

Algorithmique de base

Exercice 1. Premier pas…

Écrire un script Python traduisant l'algorithme suivant :

```
Algorithme pgcdPpcm
# cet algorithme permet de calculer le PGCD et le PPCM de deux entiers
# naturels non nuls entrés au clavier
variables   a, b, aa, bb, reste, pgcd, ppcm : entiers naturels
début
      # lecture des données
  Entrer ( a, b )
      # on veut a > b…
  si ( a < b )
  alors     aa ← a
            a ← b
            b ← aa
  fin_si
      # initialisations
  aa ← a
  bb ← b
  reste ← aa mod bb
      # boucle de calcul
  tantque ( reste ≠ 0 )
      aa ← bb
      bb ← reste
      reste ← aa mod bb
  fin_tantque
      # calcul du pgcd et du ppcm
  pgcd ← bb
  ppcm ← ( a * b ) div pgcd
      # affichage résultat
  Afficher ( pgcd, ppcm )
fin
```

Exercice 2. Lancer de pièces

Écrire un algorithme permettant de répéter `n` fois, `n` donné, la simulation de 100 lancers de pièces, et d'afficher la fréquence de l'événement « au moins 60 Piles sur 100 lancers ».

Pour utiliser le générateur de nombres aléatoires de Python, il est nécessaire d'utiliser le module `random`*. On dispose alors des primitives suivantes :*

```
import random                    # on intègre le module random
...
```

```
random.seed()                  # initialise le générateur
i = random.randrange(12)       # nombre entier aléatoire entre 0 et 11
i = random.randrange(4,12)     # nombre entier aléatoire entre 4 et 11
i = random.randrange(4,12,3)   # nombre entier aléatoire parmi 4, 7, 10
r = random.random()            # nombre flottant aléatoire entre 0.0 et
                               # 1.0 non compris (16 décimales)
```

Manipulation de listes

Exercice 3. Construction de liste

Écrire un algorithme permettant de construire une liste à partir d'entiers lus au clavier (on demandera au préalable le nombre d'éléments de la liste) et de l'afficher, dans l'ordre de sa construction, puis dans l'ordre inverse.

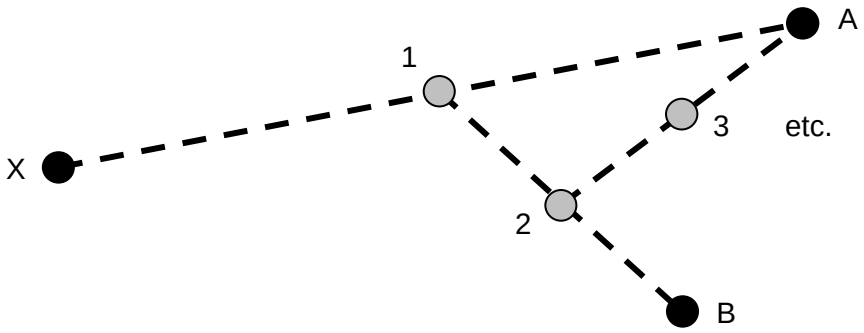
Exercice 4. Liste des diviseurs

Écrire un algorithme permettant de construire, et d'afficher, la liste des diviseurs d'un entier donné.

Aspects graphiques

Exercice 5. La flèche de Xénon (1)

Xénon est positionné en un certain point du plan X, et deux cibles sont positionnées en des points A et B. Xénon lance une flèche vers la cible A, mais celle-ci tombe au sol à la moitié de la distance souhaitée. Xénon se déplace, ramasse sa flèche, et la lance vers la cible B. Là encore, elle tombe au sol à la moitié de la distance souhaitée. Xénon continue ainsi, en visant alternativement les cibles A et B. Écrire un programme Python qui dessine la position initiale X, les points A et B, et les positions successives de Xénon (on demandera à l'utilisateur de fixer le nombre de lancers de flèche). Qu'observez-vous ?

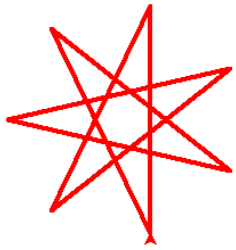


Exercice 6. La flèche de Xénon (2)

Reprendre l'exercice précédent, mais avec trois cibles A, B et C. Qu'observez-vous ?

Exercice 7. Étoile impaire

Écrire un algorithme permettant de dessiner une « étoile » à N branches (N impair), selon le modèle suivant :



Pour ceux qui auraient besoin d'exercices supplémentaires…

Exercice 8. Une suite particulière…

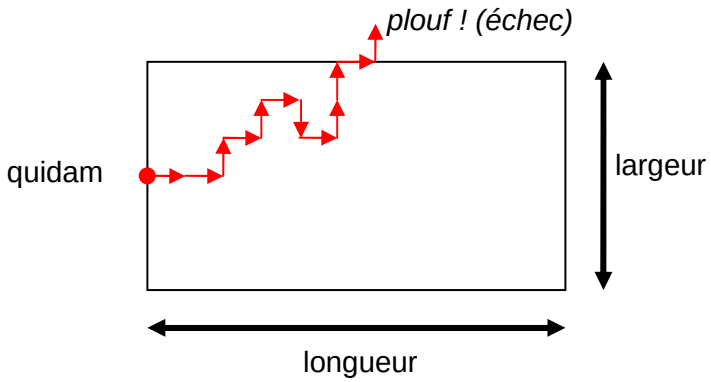
Écrire un algorithme répétant, à partir d'un entier de 4 chiffres ayant au moins deux chiffres distincts (4444 est par exemple interdit), la séquence d'opérations suivante jusqu'à ce que le résultat soit stable :

- Considérons par l'exemple l'entier N = 2672,
- On « trie » l'entier par ordre croissant de ses chiffres, on obtient C = 2267,
- puis par ordre décroissant, on obtient D = 7622,
- On calcule la différence D – C = 7622 – 2267 = 5355,
- et on recommence, 5553 – 3555 = 1998, etc.

Exercice 9. La traversée du pont

Un quidam souhaite traverser un pont de largeur et longueur donnée (en nombre de pas). Initialement, le quidam est situé au début du pont, au milieu de sa largeur. À chaque étape, aléatoirement, le quidam avance d'un pas ou se déplace d'un pas vers la gauche ou la droite.

Écrire un programma Python qui simule un nombre donné d'essais de traversée du pont et retourne la fréquence de réussite.



Exercice 10. Décomposition en facteurs premiers

Écrire une fonction permettant de décomposer un nombre en facteurs premiers. Cette fonction devra retourner une liste composée des facteurs premiers suivis de leur puissance. Par exemple, pour l'entier $1960 = 2^3 \times 5 \times 7^2$, la fonction retournera la liste [2, 3, 5, 1, 7, 2].

Utiliser cette fonction pour écrire une fonction déterminant le PGCD de deux nombres entiers positifs.

Exercice 11. Tri par insertion

Écrire une fonction permettant de rechercher par dichotomie la position d'insertion d'un élément dans une portion de liste triée (la « portion » est donnée par ses rangs de début et de fin).

Écrire une fonction permettant d'insérer un élément dans une liste en une position donnée.

Écrire une fonction permettant de trier par insertion une liste, en utilisant les deux fonctions précédentes.

Naturellement, les fonctions seront testées et validées une à une.

Exercice 12. Tri rapide

Implémentez l'algorithme de tri rapide (Quicksort) sur une liste (on définira les fonctions adéquates).