**Python**
**Exercices de révision de la période 1**

Vacances de Toussaint

---

Si vous n'avez pas installé Python et Spider sur vos ordinateurs personnels, je vous rappelle que vous pouvez télécharger la distribution ANACONDA pour Windows à l'adresse suivante : https ://www.continuum.io/downloads

les exercices proposés dans ce documents vous permettront de revoir et de mieux assimiler les notions de programmation en Python et un certain nombre de fonctions utiles en mathématiques disponibles dans les bibliothèques.

Ce travail suivant ne sera pas ramassé ni corrigé, mais en cas de difficulté, vous pouvez me contacter à l'adresse suivante : pdelahaye@cegetel.net
Je vous conseille également fortement de rester en contact les uns avec les autres afin de vous entraider.

---

## I] Le codage binaire et informatique des nombres :

1. **Codage binaire :**

   (a) Donner le codage binaire de l'entier $n = 324$.
   Vérifiez votre résultat à l'aide de la fonction `bin()`.

   (b) Proposez une procédure python nommée `codebin1()` permettant d'effectuer ce travail à votre place et utilisez-la pour vérifier votre codage précédent.
   *Bien entendu, vous éviterez de recopier bêtement le programme fait ensemble en classe.*

   (c) Retrouvez la formule permettant de déterminer le nombre de chiffres nécessaires pour coder une entier naturel $n$ en binaire.

   (d) Donner le codage binaire du décimal $d = 0.324$ avec une précision de 5 chiffres.

   (e) Proposez une procédure python nommée `codebin2()` permettant d'effectuer ce travail à votre place et utilisez-la pour vérifier votre codage précédent.
   *Bien entendu, vous éviterez de recopier bêtement le programme fait ensemble en classe.*

   (f) Expliquez pourquoi les chiffres du codage binaire de $d \in [0,\ 1[$ avec une précision de $n$ chiffres sont identiques aux chiffres du codage binaire de l'entier $N = \lfloor 2^n d \rfloor$ à condition de rajouter quelques "0" à gauche pour obtenir les $n$ chiffres.
   Vérifiez cette constatation pour obtenir le codage binaire de $d = 0.324$ à l'aide de `codebin1()`.

2. **Codage informatique :**

   (a) Déterminer le codage informatique sur 32 bits :
      i. de l'entier $n_1 = 324$.
      ii. de l'entier $n_2 = -324$.
      iii. du flottant représentant le nombre $x = 324.324$.

   (b) Quel est le nombre dont le codage informatique sur 32 bits est : 1-00101011-00010000100001100001001 ?

---

**II] SYMPY : les fonctions de calcul algébrique et d'analyse**

Pour répondre rapidement aux questions suivantes, vous utiliserez des fonctions judicieusement choisies dans les bibliothèques `math`, `sympy`, `matplotlib.pylab`, `numpy` et `scipy`.

Vous pourrez bien entendu vous aider du document qui vous a été distribué récapitulant les différentes fonctions PYTHON utiles.

Je vous rappelle que dans certains cas, une aide concernant l'utilisation de ces fonctions est disponible dans la fenêtre "explorateur" de Spyder, dès lors que la fonction a bien été importée.

1. Donner les solutions exactes complexes de l'equation $x^3 - x^2 - x - 2 = 0$.

2. Résoudre l'inéquation $x^4 + x^3 - 5x^2 + x - 6 < 0$.

3. Développer et factoriser dans $\mathbb{C}$ en facteur de degré 1, l'expression $f(x) = x(x-1) + (x^3-1)(x+2)$.

4. Donner les solutions exactes du système linéaire : $\begin{cases} x - 2y + 3z = 6 \\ 3x + y - 2z = -1 \\ -2x + 3y - z = 1 \end{cases}$.

5. Déterminer l'expression de la dérivée de la fonction $f$ définie par $f(x) = \tan^2 x$.

6. Déterminer les primitives sur $\mathbb{R}$ de la fonction $f$ définie par $f(x) = \sqrt{1+x^2}$.

7. Déterminer la forme des solutions de l'équation différentielle $y'' - 2y' + 3y = \cos(2x)$.

8. Retrouver la valeur connue de la limite en 0 de l'expression $f(x) = \dfrac{1-\cos x}{x^2}$.

9. Calculer la valeur de l'intégrale $\displaystyle\int_{-1}^{1} \arccos x \, \mathrm{d}x$.

10. Comparer les valeurs de $\pi$ obtenues, l'une avec le $\pi$ importé de `sympy`, l'autre en appliquant la formule $4 \arctan 1$ avec la fonction `atan` de la bibliothèque `math`.
    A partir de quelle décimale, ces valeurs sont distinctes ?

11. déterminer le développement limité de l'expression $f(x) = e^x$ au voisinage de 0 à l'ordre 5.

---

**III] Les fonctions de graphisme :**

1. Tracer sur $[-10,\ 10]$ le graphe de la fonction $f$ définie par $f(x) = \arcsin(\sin(x)) + \arccos(\cos(x))$.

   (a) Avec la fonction `plot()` de sympy
   (b) Avec la fonction `plot()` de matplotlib.pylab
   *Vous veillerez à importer vos fonctions depuis `matplotlib.pylab`.*

2. Sur un même graphique, tracer les droites $D_m : y = \dfrac{x}{\cos(m)} - \tan(m)$ pour tout $m \in [\![1, 100]\!]$.
   A quoi ressemble la courbe qui apparaît lorsque vous vous placez dans la fenêtre $[-5,5] \times [-5,5]$ ?

3. Une fourmi se déplace dans le plan.
   Les formules donnant ses coordonnées en fonction du temps (en secondes) sont les suivantes : $\begin{cases} x(t) = 2\cos t - \cos(2t) \\ y(t) = 2\sin t - \sin(2t) \end{cases}$.
   Représenter sur un graphe la trajectoire parcourue par la fourmi durant les 5 premières secondes.

---

## IV] SCIPY : Les fonctions de résolutions numériques

1. **Solution(s) approchée(s) d'une équation :**

   (a) Aller sur internet pour découvrir la fonction `brentq`

   (b) Utiliser cette fonction pour trouver la valeur approchée de la solution de $\cos x = x$ sur l'intervalle $[0,\ \frac{\pi}{2}]$.

   (c) A l'aide d'un graphe, identifier le nombre de racine de l'équation $x^4 - 4x^3 - 26x^2 + 60x + 50 = 0$.
   Puis utiliser la fonction `brentq` pour déterminer leur valeur approchée à $10^{-15}$ près.

2. **Calcul de** $\displaystyle\int_0^1 \frac{\tan t}{t}\,\mathrm{d}t$ **:**

   (a) Que vous renvoie la fonction `integrate()` de sympy ?

   (b) Calculer une valeur approchée de cette intégrale en utilisant les fonctions `quad()` de scipy.integrate.

   (c) Faire une recherche sur internet pour interpréter les deux valeurs renvoyées par la fonction.

3. **Résolution de l'équation différentielle** $y' = x\cos y$ **:**

   (a) Que vous renvoie la fonction `dsolve()` de sympy ?

   (b) A l'aide de la fonction `odeint()` de scipy.integrate, tracer sur $[0,\ 5]$ la solution de cette équation différentielle qui prend la valeur 0 en 0.

---

## V] Deux petits programmes :

1. **Etude de la divergence d'une somme :**

   (a) Construire une procédure d'argument $n \in \mathbb{N}$ permettant de calculer : $S(n) = \displaystyle\sum_{k=2}^{n} \frac{1}{k.\ln k}$.

   (b) Adapter la procédure précédente pour déterminer la plus petite valeur de $n$ pour laquelle $S(n)$ devient supérieure à une valeur $A$.

2. **Marche aléatoire d'une fourmi :**

   (a) La fonction `randrange()` de la bibliothèque `random` permet d'obtenir un entier naturel.
   Rechercher sur internet les modalités d'utilisation de cette fonction.

   On cherche à modéliser la marche aléatoire d'une fourmi dans le plan.
   On suppose que cette fourmi part du point $O$ et que charque seconde, elle se déplace de $p$ unités vers la droite et de $q$ unités vers le haut où $p$ et $q$ sont des entiers relatifs choisis au hasard entre $-10$ et 10. On souhaite représenter la trajectoire parcourue au bout de $N$ secondes.

   (b) Construire une liste $X$ des abscisses des points où se situe la fourmi durant les $N$ secondes

   (c) Construire une liste $Y$ des ordonnées des points où se situe la fourmi durant les $N$ secondes

   (d) En déduire la trajectoire parcourue par la fourmi durant les $N$ secondes

   (e) Faites tourner plusieurs fois votre programme et observer les différentes trajectoires obtenues.

   (f) Observer ce qui semble se passer lorsque $N$ devient très grand. Comment interpréter ce résultat ?

   (g) On s'intéresse maintenant au centre de gravité des différentes trajectoires obtenues.

i. Faire un programme simulant 100 trajectoires de 60 secondes et calculant pour chacune de ces trajectoires les coordonnées du barycentre (centre de gravité) des points constituant cette trajectoire. Les valeurs obtenues seront stockées dans deux listes `Xbar` et `Ybar`. from random import randrange from matplotlib.pylab import plot

ii. Représenter sur un graphe les différents barycentres obtenus.
Quelle figure semble apparaître lorsqu'on impose les mêmes unités sur $O_x$ et $O_y$ ?