

# Devoir Surveillé n° 1 : corrigé

PTSI Lycée Eiffel

16 novembre 2018

## Exercice 1

1. Selon ses goûts, on a l'embarras du choix. Si on préfère insister sur les aspects plus théoriques du développement de l'informatique, on citera évidemment **Alan Turing**, concepteur de la fameuse machine de Turing (modèle théorique de fonctionnement d'une machine automatique de démonstration et de calcul mathématique) puis grand contributeur du développement des premiers ordinateurs ; ou encore **Claude Shannon**, qui a étudié en détail les systèmes binaires et notamment créé le mot *bit*. Si on est plus attiré par le pognon, les noms de **Bill Gates** (fondateur de Microsoft, et pendant un temps homme le plus riche du monde) ou **Steve Jobs** (cofondateur et longuement dirigeant d'Apple, mais aussi principal actionnaire de la société Pixar à ses débuts ; sans lui, peut-être que *Toy story* n'aurait jamais vu le jour). Quelque part à la frontière entre les deux, on peut citer **Gordon Moore** (créateur de la société Intel, énonciateur de la loi qui porte son nom sur l'amélioration régulières des performances des ordinateurs).
2. C'est un logiciel qui organise le fonctionnement de l'ordinateur, de la gestion de la mémoire (explorateur de fichiers) à la répartition des tâches entre les différents logiciels fonctionnant sur l'ordinateur. Les plus utilisés à l'heure actuelle : Windows, MacOS, Linux (ces deux derniers basés sur des noyaux Unix).
3. On obtient, par exemple à l'aide de divisions euclidiennes successives,  $113 = 1110001$ , soit  $113 = 64 + 32 + 16 + 1$ .
4. Une méthode consiste à convertir 89 en binaire :  $89 = 64 + 16 + 8 + 1 = 01011001$ , à inverser tous les chiffres puis à ajouter 1 pour obtenir  $-89 = 10100111$ . Sinon, on peut aussi calculer  $256 - 89 = 167$  et le convertir directement en binaire, pour le même résultat.
5. 

```
c='miaou'
while c!='bonjour' :
    c=input('Veuillez taper le mot bonjour')
```

## Exercice 2

1. 

```
def distance(a,b) :
    return a*a+b*b
```
2. On peut importer tout le module random à l'aide de la commande : `import random`. On utilisera alors la fonction sous la forme `random.random()`. Alternativement, on n'importe que la fonction à l'aide de la commande `from random import random`, et on l'utilisera sous la forme plus simple `random()`.
3. Comme certains s'en sont rendus compte, il était beaucoup plus logique d'inverser certaines lignes du programme pour passer les définitions de *a* et *b* dans la boucle :

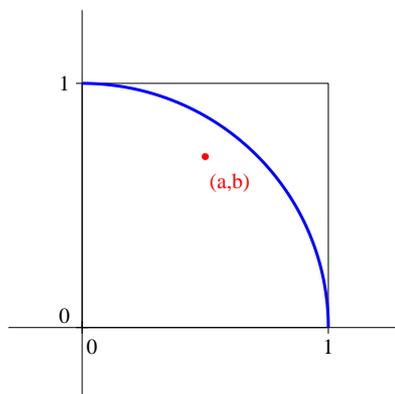
```
from random import random
def experiences(n) :
```

```

c=0
for i in range(n) :
    a=random()
    b=random()
    if distance(a,b)<=1 :
        c=c+1
return 4*c/n

```

4. Si on fait un petit dessin, on se rend compte que tirer deux nombres aléatoires (indépendants) entre 0 et 1 revient exactement à choisir au hasard un point de coordonnées  $(a, b)$  dans un carré de côté 1. Ce point vérifiera l'inégalité  $a^2 + b^2 \leq 1$  s'il est à une distance inférieure ou égale 1 de l'origine  $O$  du repère, donc s'il est situé dans le quart de cercle de rayon 1 centré en  $O$ . Or, le quart de cercle a pour aire  $\frac{\pi}{4}$  alors que le carré entier a pour aire 1. On a donc une probabilité  $\frac{\pi}{4}$  pour qu'un point aléatoire valide l'expérience. Si on répète cette expérience beaucoup de fois, on aura statistiquement une proportion de réussites égale à  $\frac{\pi}{4}$ . En multipliant par 4, on se rapprochera effectivement de  $\pi$ .



### Exercice 3

On définit une suite  $(u_n)$  par les conditions suivantes :  $u_0 = 1$  et  $\forall n \in \mathbb{N}, u_{n+1} = -\frac{u_n}{(2n+1)(2n+2)}$ .

1. 

```
def terme(n) :
    u=1
    for i in range(1,n+1) :
        u=-u/((2*n+1)*(2*n+2))
    return(u)
```
2. (a) Il aurait du mettre un return à la place du print.  
 (b) La variable  $u$  représente la valeur de  $u_n$ , ou plutôt de  $u_{n+1}$  puisque la variable a été initialisée à  $u_0$  avant même de commencer la boucle. La variable  $s$  représente la somme des termes de la suite. Plus précisément, ici, on calcule  $\sum_{k=1}^{n+1} u_k$  (le terme  $u_0$  n'est pas pris en compte dans le calcul de somme, et il y a  $n+1$  passages dans la boucle).  
 (c) Déjà il faudrait un range(n) au lieu du range(n+1) si on veut ne pas prendre en compte  $u_{n+1}$  dans le calcul de somme. Mais en fait, ne pas compter  $u_0$  est sûrement une erreur, il serait encore meilleur d'inverser les lignes 5 et 6 du programme (les modifications de  $s$  et de  $u$ ) pour mettre à jour la valeur de  $s$  avant de calculer le terme suivant de la suite. Dans

ce cas, il faut revenir à un `range(n)` pour calculer la somme des  $n + 1$  premiers termes de la suite, donc de  $u_0$  jusqu'à  $u_n$ .

3. Une méthode horrible mais efficace :

```
from math import cos, abs
n=0
while abs(miaou(n)-cos(1))>10**(-3) :
    n=n+1
print(n)
```

La méthode est horrible car ici, on recalcule à chaque fois la somme complète quand on augmente  $n$  alors qu'il suffit bien sûr d'ajouter le terme  $u_{n+1}$  pour gagner beaucoup de temps. Je laisse le lecteur motivé écrire un programme plus intelligent.