

ENSEIGNEMENT ANNUEL D'INFORMATIQUE ET D'ALGORITHMIQUE

I - Éléments d'informatique et d'algorithmique

L'objectif est de poursuivre la formation initiée au lycée concernant l'algorithmique et l'utilisation de l'outil informatique en mathématiques au travers de thèmes empruntés au programme pour comprendre, illustrer et éclairer les notions introduites. Dès qu'un calcul numérique est envisagé, dès qu'un problème incite à tester expérimentalement un résultat, dès qu'une situation aléatoire peut être modélisée avec des outils informatiques, le recours à des algorithmes et des logiciels devra devenir naturel.

L'utilisation de l'outil informatique se fait en continuité avec le cours de mathématiques et sera suivi d'une mise en œuvre sur ordinateur. Les séances de travaux pratiques doivent se faire le plus souvent possible sur ordinateur. Les étudiants, au cours de leurs études ultérieures puis de leur parcours professionnel, seront amenés à utiliser des outils informatiques divers choisis pour leurs fonctionnalités, et seule une pratique régulière de ces outils informatiques peut leur permettre d'en acquérir la maîtrise. De plus, en adoptant cette démarche exploratoire permise par le dialogue interactif avec la machine, cette pratique peut s'avérer bénéfique pour les apprentissages et faciliter la compréhension de concepts plus abstraits.

Le programme d'informatique s'articule autour de trois thèmes : études de suites , statistiques descriptives univariées, bases de données relationnelles.

L'ordre dans lequel les thèmes sont abordés est libre, mais il est préférable de mener ces activités en cohérence avec la progression du cours de mathématiques.

Les exemples traités dans un thème devront être tirés, autant que possible, de situations réelles (traitement de données économiques, sociologiques, historiques, démographiques, en lien avec le monde de l'entreprise ou de la finance, etc.), en faisant dès que possible un rapprochement avec les autres disciplines.

Pour certains thèmes, il sera nécessaire d'introduire de nouvelles notions mathématiques ; celles-ci seront introduites en préambule lors des séances d'informatique ; elles ne pourront en aucun cas être exigibles des étudiants, et toutes les précisions nécessaires seront données lors de leur utilisation.

Le langage informatique retenu pour la programmation dans ce programme des classes économiques et commerciales, option technologique, est Python.

1 - Langage Python

Le langage Python propose un grand nombre de bibliothèques logicielles, avec des utilités variées. Les bibliothèques jugées nécessaires sont listées, chacune avec une liste restreinte de fonctions essentielles que les étudiants devront avoir manipulées. Seules celles dans la colonne de gauche sont exigibles, et leur syntaxe précise doit être rappelée. D'autres fonctions, par commodité, pourront être utilisées en classe, mais ceci ne pourra se faire qu'avec parcimonie. L'objectif principal de l'activité informatique reste la mise en pratique de connaissances mathématiques.

a) Types de base

Affectation : `nom = expression`

L'expression peut être du type numérique, booléen, matriciel (ndarray) ou chaîne de caractères.

permet d'insérer un commentaire

+	-	*	/	**
---	---	---	---	----

==	>	<	>=	<=	!=
----	---	---	----	----	----

True	False	and	or	not
------	-------	-----	----	-----

from ... import *, import ... as

On insiste sur l'importance de l'ajout judicieux de commentaires.

Opérations arithmétiques de base.

Comparaison, test.

Logique.

Importation d'une bibliothèque.

b) Structures de contrôle

On réinvestit les notions de compteurs et d'accumulateurs vues au lycée. La maîtrise des structures de programmation de base (if, while, for) constitue l'un des objectifs majeurs de l'informatique en première année.

Instruction d'affectation : =.

Instruction conditionnelle if, elif, else.

Boucle for; Boucle while.

Définition d'une fonction :

def f(p₁, ... , p_n)

return.

c) Utilisation de bibliothèques

Pour le calcul numérique, le traitement statistique ou la simulation de phénomènes aléatoires, certaines bibliothèques s'avèrent utiles. Elles sont listées ci-dessous avec les fonctions pertinentes. Toute utilisation d'une telle fonction doit obligatoirement être accompagnée de la documentation utile, sans que puisse être attendue une quelconque maîtrise par les étudiants de ces éléments.

from ... import *, import ... as

Importation d'une bibliothèque.

— Dans la bibliothèque `numpy`

Exemple d'importation : `import numpy as np`

`np.e`, `np.pi`

`np.exp`, `np.log`, `np.sqrt`, `np.abs`,

`np.floor`

`np.array`.

`np.dot`

+	-	*	/	**
---	---	---	---	----

`np.sum`, `np.min`, `np.max`, `np.mean`,

`np.cumsum`, `np.median`, `np.var`, `np.std`

Constantes e et π.

Ces fonctions peuvent s'appliquer à des variables numériques ou vectoriellement (à des matrices ou vecteurs) élément par élément. Création de tableaux ou matrices.

`np.zeros`, `np.ones`, `np.eye`, `np.arange`, `np.linspace`, `np.reshape`.

Extraction ou modification d'un élément, d'une ligne ou d'une colonne du tableau.

Multiplication matricielle.

On pourra utiliser `a, b = np.shape(M)` pour obtenir la taille de la matrice M.

Opérations arithmétiques de base : coefficient par coefficient.

Ces opérations peuvent s'appliquer sur une matrice entière ou bien pour chaque colonne (ou chaque ligne). Exemple : `mean(M)`, `mean(M,0)`, `mean(M,1)`

- Dans la librairie `numpy.random`
Exemple d'importation : `import numpy.random as rd`
`rd.random.`
- Dans la librairie `pandas`
Exemple d'importation : `import pandas as pd`
`pd.mean, pd.std.` Indicateurs statistiques.
- Dans la librairie `matplotlib.pyplot`
Exemple d'importation : `import matplotlib.pyplot as plt`
`plt.plot, plt.show` Représentations graphiques de fonctions, de suites. On pourra utiliser les commandes non exigibles `xlim, ylim, axis, grid, legend`.
Représentations statistiques
`plt.hist, plt.bar, plt.boxplot.`

2 - Liste de savoir-faire exigibles en première année

C1 : Savoir manipuler les structures algorithmiques de base (if, for, while), connaître la syntaxe d'une fonction simple et savoir l'utiliser.

C2 : Savoir produire des graphiques et indicateurs afin d'interpréter des données statistiques.

C3 : Savoir étudier des suites numériques, calculer des valeurs, tracer des graphiques et conjecturer des résultats sur le comportement de la suite.

C4 : Stocker, organiser et extraire des données structurées volumineuses.

C5 : Savoir modéliser des phénomènes aléatoires et effectuer des simulations de variables aléatoires.

II - Liste de thèmes

1 - Suites

(Durée indicative : 6 heures. Compétences développées : **C1** et **C3**)

Calcul des termes d'une suite.

Exploitation graphique des résultats.

Exemples : taux d'intérêt, emprunt.

Calculs de valeurs approchées de la limite d'une suite.

On utilisera des structures répétitives et conditionnelles en exploitant l'étude mathématique.

Valeur approchée d'une constante.

Détermination du rang d'arrêt.

2 - Statistiques descriptives univariées

(Durée indicative : 3 heures. Compétences développées : **C2**)

Dans ce paragraphe, on analysera des données statistiques issues de l'économie, du monde de l'entreprise ou de la finance, en insistant sur les représentations graphiques. On insistera sur le rôle des différents indicateurs de position et de dispersion étudiés.

Série statistique associée à un échantillon.
 Effectifs, fréquences, fréquences cumulées, diagrammes en bâton, histogrammes.
 Indicateurs de position : moyenne, médiane, mode, quantiles.
 Indicateurs de dispersion : étendue, variance et écart-type empiriques, écart inter-quantile.
 Analyse d'un caractère quantitatif : caractéristiques de position (moyenne, médiane); mode(s); caractéristiques de dispersion (variance et écart-type empiriques, quartiles, déciles).

On pourra également utiliser les commandes :
`pd.read_csv`, `head`, `shape`, `pd.describe`

`pd.median`, `pd.count`, `pd.sort_values`

On notera bien que les paramètres empiriques sont calculés à partir de l'échantillon observé. On montrera les avantages et les inconvénients des caractéristiques liées à la structure euclidienne (moyenne et écart-type) et ceux qui sont liés à la structure d'ordre (quantiles).

3 - Bases de données

(Durée indicative : 6 heures. Compétences développées : **C4**)

L'administration, les banques, les assurances, les secteurs de la finance utilisent des bases de données, systèmes d'informations qui stockent dans des fichiers les données nombreuses qui leur sont nécessaires. Une base de données relationnelle permet d'organiser, de stocker, de mettre à jour et d'interroger des données structurées volumineuses utilisées simultanément par différents programmes ou différents utilisateurs. Un logiciel, le système de gestion de bases de données (SGBD), est utilisé pour la gestion (lecture, écriture, cohérence, actualisation...) des fichiers dans lesquels sont stockées les données. L'accès aux données d'une base de données relationnelle s'effectue en utilisant un langage informatique qui permet de sélectionner des données spécifiées par des formules de logique, appelées requêtes d'interrogation et de mise à jour.

L'objectif est de présenter une description applicative des bases de données en langage de requêtes SQL (Structured Query Language). Il s'agit de permettre d'interroger une base présentant des données à travers plusieurs relations. On pourra pour introduire la problématique donner l'exemple de la base de données utilisée par un progiciel de gestion intégré (PGI), outil informatique permettant de piloter une entreprise, présenté dans le cours de management. Cette base de données stocke les informations communes à de nombreux services (comptabilité, gestion du personnel, gestion des stocks, fichier clients...). On introduira les concepts d'interrogation et de mise à jour d'une base de données à l'aide d'exemples simples issus de ce contexte.

Modèle relationnel : relation, attribut, domaine, clef primaire "PRIMARY KEY", clef étrangère "FOREIGN KEY", schéma relationnel.

Vocabulaire des bases de données : table, champ, colonne, schéma de tables, enregistrements ou lignes, types de données.
 Lecture d'un fichier de données simples. Notion de descripteur.

Opérateurs arithmétiques +, -, *.

Opérateurs de comparaison :
 =, <>, <, <=, >, >=.

Opérateurs logiques : "AND", "OR", "NOT".

On s'en tient à une notion sommaire de domaine : entier "INTEGER", chaîne "TEXT".

a) Commandes exigibles

"WHERE"

"SELECT nom_de_champ FROM nom_de_table". Sélection de données dans une table.

"INSERT INTO nom_de_table ". Insertion de données dans une table. On pourra utiliser "VALUES (élément1, élément2,...)".

"DELETE FROM nom_de_table ". Suppression de données d'une table.

"UPDATE nom_de_table ". Mise à jour de données d'une table.

b) Commandes non exigibles

On pourra utiliser par commodité et si besoin la liste d'opérateurs, fonctions et commandes ci-dessous. Ce ne sont pas des attendus du programme et ils sont non exigibles.

Les opérateurs ensemblistes : union "UNION", intersection "INTERSECTION", différence "EXCEPT".

Les opérateurs spécifiques de l'algèbre relationnelle : projection, sélection (ou restriction), renommage, produit cartésien .

Les fonctions d'agrégation : min "MIN", max "MAX", somme "SUM", moyenne "AVG", comptage "COUNT".

Les commandes "DISTINCT", "ORDER BY"

4 - Probabilités

(Durée indicative : 3 heures. Compétences développées : **C1 C2 et C5**)

Utilisation de la fonction `rd.random` pour simuler des expériences aléatoires élémentaires conduisant à une loi usuelle. Loi uniforme, loi binomiale. `rd.randint` .

Simulation de phénomènes aléatoires.