CHAPITRE 5
Données et traitements

Introduction

La thématique des données est celle qui fait le plus appel à la programmation. En effet, algorithmique et programmation sont au cœur des enjeux de la science des données et du « *big data* ». Ce chapitre propose des activités donnant une large place aux données ouvertes et aménage une transition en douceur de l'analyse de données avec un tableur à la programmation de cette analyse.
L'enjeu est de donner aux lycéens les moyens d'analyser les données disponibles dans les domaines culturels, économiques ou sociaux qui les intéressent. Les enjeux environnementaux des « *data centers* » et la question des données personnelles occupent une place importante dans les pistes de réflexion et de débats proposés.

Activités de découverte

Hormis la première activité autonome, toutes les activités de découverte proposées sont en mode débranché. Le traitement de données nécessite en effet de réfléchir aux algorithmes avant de se lancer dans la programmation. Cette réflexion peut être menée à partir d'exemples de traitement de données tirés de la vie quotidienne ou de travaux scolaires menés dans d'autres disciplines.

Activité 1. Un site à explorer : data.gouv.fr

- Activité **branchée** nécessitant une connexion à internet : exploration d'un site emblématique du thème étudié.
- Adresse du site : https://www.data.gouv.fr

L'activité de découverte autonome a pour objectif de montrer aux élèves l'étendue des données ouvertes disponibles dans tous les domaines d'activité. Il s'agit donc d'éveiller leur curiosité et de montrer que des informations sont disponibles et ne demandent qu'à être analysées.
L'objectif secondaire est aussi de faire prendre conscience que l'accès aux données n'est pas toujours simple car il faut comprendre le format des données obtenues. Pour simplifier l'activité, on ne s'intéresse qu'aux fichiers de type csv, qui posent déjà la question du choix du séparateur utilisé.
L'activité est proposée à partir du site national **data.gouv.fr** mais peut être transposée sans difficulté à un autre site de données ouvertes si une collectivité locale (région, ville…) en propose.

Activité 2. Recherches dans un annuaire papier

- Activité **débranchée**.
- Durée estimée : 15 à 30 minutes.

Même si les annuaires papier tendent à disparaître, l'activité proposée renvoie à un quotidien vécu par les élèves. Il n'est pas nécessaire de disposer d'un annuaire par

élève pour réaliser cette activité (attention au poids des cartables !). L'objectif est de faire réfléchir à la méthode employée pour effectuer différentes recherches dans un annuaire. Il suffit donc de disposer d'un annuaire en consultation pour savoir dans quel ordre sont rangées les informations.

1. Rechercher un boulanger dans une commune donnée nécessite d'abord de rechercher la profession puis, à l'intérieur des pages consacrées à cette profession, de rechercher la commune, puis à l'intérieur de cette commune, de rechercher le nom du professionnel. Pour mener à bien cette recherche rapidement, il suffit de savoir dans quel ordre sont rangées les informations dans un annuaire *Pages jaunes*.

2. Rechercher la liste de tous les professionnels d'une petite commune d'un département est une tâche très fastidieuse, car il faut regarder successivement dans toutes les professions si cette commune est représentée.

3. Ce défi peut être mis en place par binôme : un élève note un numéro figurant dans l'annuaire et propose à son binôme de rechercher le nom du professionnel en lui donnant seulement le numéro.
Il est inutile de mener l'activité jusqu'au bout pour se rendre compte du temps nécessaire pour parcourir ligne à ligne l'annuaire à la recherche du numéro !

4. Réaliser un annuaire papier inversé, outre que cette fabrication gaspillerait beaucoup d'arbres, n'a plus beaucoup de sens à l'époque des annuaires inversés en ligne. L'objectif de cette question est simplement de faire réfléchir à l'ordre de classement qu'il faudrait utiliser pour que la recherche par numéro soit possible.
La réponse est évidemment de trier les numéros par ordre croissant pour permettre un accès rapide comme dans un dictionnaire.

5. Le classement adopté par les annuaires de professionnels est généralement par profession, puis par commune, puis par nom. Cela facilite le type de recherche de la question **1**. Pour favoriser une recherche du type de celle de la question **2**, il faudrait que l'annuaire soit d'abord trié par communes puis, à l'intérieur de chaque commune, par profession et, finalement, à l'intérieur de chaque profession par nom.

Cette activité permet ainsi de sensibiliser les élèves à l'impact du rangement des données sur leur facilité d'accès. La principale différence entre des données imprimées et des données sur ordinateur vient du fait que pour des données imprimées, l'ordre de classement est fixé à l'impression. Avec des données sur ordinateur, le tri peut être effectué à tout moment sur différents critères.
L'importance des tris en informatique reste fondamentale pour accélérer l'accès aux données. Sachant que trier un fichier de plusieurs millions d'enregistrements peut prendre un certain temps, il reste utile de réfléchir au préalable sur le classement des données pour favoriser une recherche particulière.

Activité 3. Réaliser un histogramme

- Activité **débranchée**.
- **Durée estimée : 30 minutes.**

La réalisation d'un histogramme est une activité proposée dans plusieurs disciplines scolaires, en mathématiques mais aussi en sciences de la vie ou en sciences économiques et sociales. Dans tous les cas, il s'agit d'un traitement de données où l'objectif est de remplacer une grande collection de données par un comptage de ces données réparties en plusieurs classes, en vue d'en faire une représentation graphique. On s'intéresse dans cette activité uniquement au remplissage du tableau de comptage.

1. Selon la rapidité des élèves, parcourir l'ensemble des 123 données du tableau en recherchant combien sont comprises entre 18 et 21 peut prendre entre 45 secondes et une minute. Il faut alors recommencer pour la tranche d'âge suivante, etc.
Pour compter les dix tranches d'âge, on peut donc estimer le temps de travail à environ 10 minutes.

2. Avec le second algorithme, on parcourt une seule fois les données en notant à chaque donnée un bâton dans la tranche correspondante. Ce traitement prend au plus cinq minutes, auquel il faut ajouter au plus une minute pour compter les bâtons de chaque tranche, ce travail étant facilité si on a pris soin de regrouper les bâtons par paquets de 5.

3. La comparaison devrait mettre en évidence la plus grande rapidité du 2nd algorithme. Cela s'explique simplement par le nombre de fois où l'on consulte une donnée pour la comparer aux bornes des tranches d'âge. Dans le premier algorithme, on effectue l'opération 123 fois puis on recommence ce traitement 10 fois. Dans le second, on ne regarde chaque donnée qu'une seule fois. L'opération pour chacune est un peu plus compliquée, car il faut trouver directement la bonne tranche d'âge et noter un bâton, mais cette opération n'est répétée que 123 fois.

Cette activité permet de montrer qu'il peut y avoir plusieurs méthodes plus ou moins rapides pour traiter des données. L'étude des algorithmes, ou « algorithmique » est d'ailleurs un des sujets importants en informatique, que l'on peut mentionner pour information, mais qu'il n'est pas question d'aborder en classe de seconde.

Activité 4. Le fichier bristol de la bibliothèque

- Activité **débranchée**.
- **Durée estimée : 30 minutes.**

Le travail du bibliothécaire repose sur l'utilisation de méthodes systématiques – des algorithmes – pour bien retrouver les fiches des ouvrages, celles des lecteurs et bien noter les emprunts en cours.
Il ne s'agit pas ici de formaliser ces méthodes, mais de les décrire assez précisément pour les expliquer à un interlocuteur humain.

1. Quand un lecteur souhaite emprunter un ouvrage, il faut d'abord lui demander son nom, puis le titre et l'auteur de l'ouvrage. Avec le nom de l'auteur, on peut rechercher la fiche ouvrage puis, avec le nom du lecteur, on peut rechercher sa fiche.
Il ne reste plus qu'à insérer la fiche ouvrage à côté de la fiche lecteur. Toutes ces opérations sont rapides car les fichiers sont bien triés.

2. Quand un lecteur rend un ouvrage, il faut lui demander son nom. On peut alors rechercher sa fiche dans le fichier lecteur. On doit alors trouver la fiche ouvrage stockée à côté de la fiche lecteur.
Il ne reste plus qu'à ranger la fiche ouvrage dans le fichier ouvrage, en insérant cette fiche au bon endroit selon l'ordre alphabétique.

3. Si un lecteur demande un ouvrage par son titre alors que les fiches ouvrages sont triées par ordre alphabétique d'auteur, deux cas se présentent.
Soit le bibliothécaire connaît le nom de l'auteur, cela ramène à la question **1**.
Soit le bibliothécaire ne connaît pas l'ouvrage, alors la seule solution à sa disposition est de parcourir toutes les fiches ouvrage à la recherche de celle dont le titre est le titre recherché.

4. Selon le nombre d'ouvrages de la bibliothèque, cela peut prendre un certain temps de regarder toutes les fiches ouvrages à la recherche d'un titre particulier.

5. Pour compter le nombre total de livres en distinguant ceux qui sont empruntés, on peut d'une part parcourir toutes les fiches ouvrages du fichier ouvrage en comptant le nombre de fiches, et d'autre part compter combien de fiches ouvrages ont été placées dans le fichier lecteurs pour marquer que les ouvrages correspondant ont été empruntés.

Cours

- Le cours permet de poser progressivement les notions importantes. La notion de table en tant que collection de données est utilisée dans d'autres disciplines pour noter des informations structurées de manière régulière avec des « critères ».
- Pour la mise en œuvre informatique avec des fichiers, on privilégie le format csv qui a l'avantage d'être d'usage courant, d'être le format le plus élémentaire pour représenter des tables et d'être lisible à la fois avec un tableur et par programmation.
- Concernant les traitements de données étudiés, on concentre l'étude sur les traitements pouvant être effectués au fur et à mesure de la lecture du fichier, ce qui permet d'obtenir toujours le même schéma d'algorithme.
  Les traitements plus complexes permettant de trier ou de croiser des données sont reportés dans la partie « Pour aller plus loin ».

Activités d'application

Les activités permettent de passer progressivement d'une méthode de traitement de données générale avec un tableur à une méthode de traitement spécifique définie par le programmeur.

Activité 5. La population en Europe

- Activité **branchée** sur poste informatique.
- **Durée estimée : 15 à 30 minutes.**
- **Fichiers élève et corrigé disponibles sur site compagnon et Bibliomanuel : Poste informatique. Chap5_Population.csv ; Chap5_Population_Solution.ods**

Cette activité consiste à effectuer de manière élémentaire, avec un tableur, des calculs sur une table contenant à la fois des données de population et de superficie pour les pays européens.

1. L'ouverture d'un fichier csv avec un logiciel tableur est géré différemment selon les logiciels. La suggestion d'ouvrir d'abord le fichier avec un éditeur de texte – type Notepad – est une consigne de prudence, pour identifier de manière sûre le séparateur de champ utilisé (ici le point-virgule).

2. Avec le module **Calc** de LibreOffice, l'ouverture d'un csv provoque systématiquement l'ouverture d'une boite de dialogue permettant d'indiquer le séparateur.

3. L'ajout de la colonne `Densité` nécessite d'introduire dans chaque case une formule permettant de calculer la densité en hab/km^2 en fonction de la population en nombre d'habitants et de la superficie (donnée en milliers de km^2).
La première densité peut être calculée en saisissant dans la cellule **D2** la formule suivante : `=B2/(C2*1000)`.

Les formules suivantes peuvent être générées par recopie vers le bas, si la formule initiale a bien été notée de manière relative.

4. La somme peut être calculée avec la fonction **Somme** en sélectionnant la plage de données pertinente. Un corrigé de l'activité est disponible dans le fichier Chap5_Population_Solution.ods proposé en téléchargement.

Activité 6. La résistance des pièces

- Activité **branchée** sur poste informatique.
- Durée estimée : 15 à 30 minutes.
- Fichiers tableur élève et corrigé disponible sur site compagnon et Bibliomanuel :
  Poste informatique. Chap5_Mesures.csv ; Chap5_Mesures_Solution.ods

Cette activité consiste à étudier un tableau de mesures. Elle est volontairement très proche de l'**activité 5**. L'objectif est bien de montrer la régularité que l'on peut trouver dans des traitements de tables élémentaires, y compris dans des domaines d'application différents.

1. L'ouverture préalable du fichier doit permettre d'identifier qu'il y a deux symboles de ponctuation différents qui sont utilisés. La virgule est utilisée pour noter des nombres décimaux alors que le point-virgule est bien utilisé comme séparateur de champs.

2. L'ouverture du fichier dans un tableur avec séparateur point-virgule permet de séparer correctement les colonnes de données.

3. L'ajout de la colonne Résistance amène à saisir une formule. La valeur de la résistance (en ohms) est égale au quotient de la tension (en volts) par l'intensité (en ampères).

4. Lors de l'ajout du graphique, il convient de bien sélectionner toutes les données en incluant les intitulés de lignes et de colonnes, puis de signaler à l'aide des boites de dialogues spécifiques que la première ligne et la première colonne contiennent des intitulés.
Un corrigé de l'activité est disponible dans le fichier Chap5_Mesures_Solution.ods.

Activité 7. Densités de la population en Europe

- Activité **branchée** sur poste informatique.
- Durée estimée : 15 à 30 minutes.
- Fichiers tableur et Python élèves et corrigés disponible sur site compagnon et Bibliomanuel :
  Poste informatique. Chap5_Population.csv ; Chap5_PopulationTotale.py ; Chap5_Population_Solution.py

L'objectif de cette activité est de refaire le traitement de l'activité **5** en le transposant en programmation Python.

1. Le programme fourni et le fichier csv doivent être dans le même dossier. Avec l'éditeur Mu, le plus simple est de sauvegarder tous les programmes et les données dans le dossier par défaut de l'éditeur. À l'exécution, le programme doit afficher : `Population totale : 508450856`.

2. Le calcul et l'affichage de la superficie totale de l'Europe nécessite l'ajout dans la boucle d'une instruction de calcul : `supEurope = supEurope + float(sup)`
puis à la fin du programme, l'ajout d'une instruction d'affichage :
`print("Superficie totale :", supEurope, "milliers de km2")`.

3. Le calcul et l'affichage de la densité de population peut se faire au cours de la boucle par l'ajout de l'instruction :
`print(nom, ":", int(pop)/(float(sup)*1000), "habitants/km2").`
Un corrigé de l'activité est disponible dans le fichier Chap5_Population_Solution.py.

Activité 8. Un placement rentable

- Activité **branchée** sur poste informatique.
- Durée estimée : 30 minutes.
- Fichiers corrigés disponibles sur site compagnon et Bibliomanuel :
  Poste informatique. Chap5_Credit_Solution.ods ; Chap5_Credit_Solution.py

Cette activité est une activité mixte à faire à la fois avec un tableur et en programmation Python.
L'objectif est de montrer le lien entre un calcul répétitif effectué avec un tableur en déroulant les calculs répétitifs ligne après ligne, et le même calcul programmé en Python avec une instruction répétitive.
Il faut noter dans cette activité la différence de retour pour l'élève sur ce qu'il fait calculer à l'ordinateur.
Dans la version tableur, tous les calculs intermédiaires sont visibles dans les différentes cellules. Dans la version programmée, ne sont visibles que les résultats dont le programmeur a demandé l'affichage.
Les valeurs intermédiaires peuvent être visualisées grâce à l'outil de mise au point en exécution pas à pas.

Partie 1. Simulation avec un tableur

1. Le tableau de comparaison entre les deux placements peut avoir la forme suivante : des formules permettent de calculer en 2e colonne les intérêts cumulés `=B2*1,005` et en 3e colonne les intérêts fixes `=C2+1`.

2. Ce n'est qu'à partir de la 253e période que le placement de type ① devient toujours plus avantageux. Une analyse trop rapide peut amener des élèves à conclure de manière erronée que le placement ② serait plus avantageux, ce qui n'est vrai qu'au début.
Un corrigé de cette partie est disponible dans le fichier Chap5_Credit_Solution.ods.

Partie 2. Exécution en Python

1. En suivant l'indication de calculer aussi longtemps que le placement ① est moins avantageux, le programme Python suivant permet de calculer directement la solution.

2. L'exécution du programme s'arrête après 253 périodes quand le placement 1 devient supérieur au placement ② en affichant :
`Apres 253 mois +1% donne : 353.19424979227233 et +1 donne : 353`
Le programme est disponible dans le fichier Chap5_Credit_Solution.py.

```
m1 = 100
m2 = 100
n = 0
while m1 <= m2:
    m1 *= 1.005
    m2 += 1
    n += 1
print("Apres", n, "mois +1% donne :", m1, "et +1 donne :", m2)
```

Cette activité montre que le même problème peut être résolu de manière assez différente en utilisant des outils informatiques différents. La solution avec le tableur peut

sembler plus explicite au départ. La solution programmée sera finalement beaucoup plus efficace pour aboutir directement au résultat.

La tâche de l'élève est ici de s'abstraire petit à petit des résultats intermédiaires, en construisant une représentation mentale de leur calcul, pour aboutir finalement à une solution ne donnant que le résultat final en fonction des données du problème.

Activité 9. Visualisation de données avec Panda 

- Activité **branchée** sur poste informatique.
- Durée estimée : 15 minutes.
- Fichiers corrigés disponibles sur site compagnon et Bibliomanuel :
  Poste informatique. Chap5_indicesSaintDenis.csv ; Chap5_PandaSaintDenis.py

Cette activité est une activité d'ouverture vers la science des données pour montrer comment l'usage d'une bibliothèque adaptée peut simplifier l'écriture des traitements. Le principal inconvénient de ce genre de bibliothèque est de masquer la complexité des traitements effectués. D'un point de vue pédagogique, il est aussi plus formateur de reprogrammer un traitement pour en comprendre les différentes étapes. L'existence de cette activité est justifiée par les perspectives que donnent l'usage de bibliothèques élaborées. C'est une approche complémentaire de l'approche principale choisie dans ce chapitre où les traitements sont le plus souvent expliqués de manière élémentaire.

1. Pour que cette activité soit possible, il est nécessaire que la bibliothèque **Panda** soit installée.

2. L'exécution produit un graphique montrant l'évolution annuelle du paramètre de pollution choisi. Il n'est pas utile de rentrer dans le détail de la présentation des structures de données sous-jacentes à la bibliothèque. Il suffit d'admettre que la lecture du csv charge tout le contenu du tableau dans la variable **sd**. La notation `sdpm10` permet d'accéder directement au contenu de la colonne dont `pm10` est l'intitulé.

3. La modification du programme peut se faire simplement par imitation, en remplaçant le nom du champ étudié par un autre. Les activités suivantes permettent de réaliser sur ces mêmes données des traitements plus spécifiques en les programmant de manière plus détaillée et élémentaire.

Activité 10. La qualité de l'air en Île-de-France 

- Activité **branchée** sur poste informatique.
- Durée estimée : 30 minutes.
- Fichiers corrigés disponibles sur site compagnon et Bibliomanuel :
  Poste informatique. Chap5_indices_IDF_2017.csv ; Chap5_NO2_SaintDenis.py ; Chap5_Indices_SaintDenis_Solution.py

Les activités **10** à **12** de ce chapitre portent sur le même jeu de données. Pour permettre de les réaliser de manière indépendante, les données intermédiaires sont fournies. Cette activité est guidée pour permettre à tous les élèves de dépasser facilement les obstacles techniques liés à la manipulation de fichiers depuis un programme et au traitement des chaines de caractères à décomposer pour retrouver les valeurs de chaque champ.

1. Cette question d'observation permet de constater ce que fait l'exécution du programme dans sa version initiale. Il est utile de faire remarquer que cette exécution est quasi instantanée alors que le fichier analysé comporte 476 486 lignes.

2. La tâche demandée de modification du programme a pour objectif de faire lire attentivement le programme donné pour identifier la partie du calcul portant sur le taux de dioxyde d'azote NO_2, et en déduire les modifications à apporter pour calculer les autres indices.

3. Les lycéens d'Île-de-France seront favorisés dans cette activité, car ils pourront simplement en modifiant le code INSEE exécuter le programme d'analyse de la qualité de l'air pour leur commune.

Le programme complet (pour la commune de Saint-Denis) est alors le suivant (en gras les lignes ajoutées pour la question **2**) :

```
f = open("Chap5_indices_IDF_2017.csv", "r")
entete = f.readline()
s_no2,s_o3,s_pm10 = 0, 0, 0
n = 0
ligne = f.readline()
while ligne != "":
    date,ninsee,no2,o3,pm10 = ligne.split(",")
    if ninsee == '93066':
        n += 1
        s_no2 += int(no2)
        s_o3 += int(o3)
        s_pm10 += int(pm10)
    ligne = f.readline()
print("Moyenne annuelle indice no2 :", s_no2 / n)
print("Moyenne annuelle indice o3 :", s_o3 / n)
print("Moyenne annuelle indice pm10 :", s_pm10 / n)
f.close()
```

Ce programme est une application directe de l'algorithme général de traitement de fichier (voir manuel élève page 97).

Activité 11. La qualité de l'air dans une commune

- Activité **branchée** sur poste informatique.
- Durée estimée : 30 minutes.
- Fichiers corrigés disponibles sur site compagnon et Bibliomanuel : Chap5_indices_IDF_2017.csv ; Chap5_NO2_SaintDenis.py ; Chap5_FiltreSaintDenis_Solution.py

La difficulté de cette activité réside dans le fait de manipuler deux fichiers dans le même programme : un fichier de données à lire et un extrait du premier fichier fabriqué par ce programme.
Il convient d'être vigilant pour ne pas mélanger les deux traitements sachant que les instructions se ressemblent. En particulier, l'instruction d'ouverture d'un fichier en lecture ne diffère de l'ouverture en écriture que par le caractère « r » (pour *read* : lecture) ou « w » (pour *write* : écriture).

1. Il est proposé de partir d'un programme existant, pour ne pas tout avoir à réécrire. En effet, le filtrage dans un fichier des informations concernant une commune nécessite exactement les mêmes traitements pour lire ligne à ligne le fichier d'origine.

2. Seul change la partie du traitement, où l'on remplace le calcul de la somme des indices par la réécriture dans un nouveau fichier de la ligne lue quand elle concerne la commune choisie.

3. Pour écrire la ligne d'entête dans le nouveau fichier, il suffit d'identifier l'instruction qui a permis de lire cet entête dans le fichier d'origine.

4. À l'exécution, un nouveau fichier doit apparaître dans le dossier où a été exécuté le programme et où figurait le fichier de données.

5. L'ouverture du fichier résultat permet de vérifier que son contenu est celui prévu, à savoir l'ensemble des relevés journaliers uniquement pour la commune choisie. Il doit donc contenir au maximum 366 lignes.

Le programme Python solution est le suivant et est disponible dans le fichier Chap5_FiltreSaintDenis_Solution.py.

```
f1 = open("Chap5_indices_IDF_2017.csv", "r")
f2 = open("indicesSaintDenis.csv", "w")
entete = f1.readline()
f2.write(entete)
ligne = f1.readline()
while ligne != "":
    date,ninsee,no2,o3,pm10 = ligne.split(",")
    if ninsee == '93066':
        f2.write(ligne)
    ligne = f1.readline()
f1.close()
f2.close()
```

Ce programme filtre bien dans le fichier initial contenant tous les relevés quotidiens d'Île-de-France, uniquement les relevés concernant la commune de Saint-Denis pour les réécrire dans un nouveau fichier nommé indicesSaintDenis.csv.

Activité 12. Histogramme de pollution

- Activité **branchée** sur poste informatique.
- Durée estimée : 30 à 45 minutes.
- Fichiers corrigés disponibles sur site compagnon et Bibliomanuel : Chap5_indicesSaintDenis.csv ; Chap5_Act12_Solution_Question1.py ; Chap5_Act12_Solution_Question2.py ; Chap5_Act12_Solution_Question3.py

Le fichier de données de cette activité peut être soit celui fourni pour la commune de Saint-Denis soit celui généré lors de l'activité **11** pour une commune au choix des élèves. L'activité est progressive et permet de compléter le programme au fur et à mesure.

1. Le programme peut être construit en reprenant le programme donné à l'activité **10** à condition de bien modifier le nom du fichier à ouvrir en lecture. Il n'y a plus de condition à écrire sur le code INSEE, puisque le nouveau fichier de données ne contient que les relevés d'une commune. Il faut à la place écrire une condition permettant de tester le dépassement du seuil de pollution élevée aux particules fines.

2. La règle indiquée se traduit par une nouvelle condition portant sur les trois indices de pollution.

3. Le comptage des jours où la qualité de l'air a été très faible, faible, moyenne, élevée ou très élevée nécessite l'usage d'autant de variables pour mémoriser ces nombres au fur et à mesure de la lecture du fichier.

4. L'affichage du résultat sous forme de tableau nécessite de formater précisément les valeurs avec un nombre de caractères précis, pour que les alignements soient corrects. Les informations techniques peuvent être données aux élèves pour éviter toute difficulté technique inutile.

Les corrigés des différentes étapes sont disponibles dans les fichiers : Chap5_Act12_Solution_Question1.py ; Chap5_Act12_Solution_Question2.py ; Chap5_Act12_Solution_Question3.py

Le programme final est le suivant :

```
f = open("Chap5_indicesSaintDenis.csv", "r")
entete = f.readline()
ligne = f.readline()
ntf, nf, nm, ne, nte = 0, 0, 0, 0, 0
while ligne != "":
    date,ninsee,no2,o3,pm10 = ligne.split(",")
    indice = max(int(no2), int(o3), int(pm10))
    if indice < 25:
        ntf = ntf+1
    elif indice < 50:
        nf = nf+1
    elif indice < 75:
        nm = nm+1
    elif indice <= 100:
        ne = ne+1
    else:
        nte = nte+1
    ligne = f.readline()
print("Pollution")
print("|Très Faible| Faible | Moyenne | Elevée |Très élevée|")
print("| {:6} | {:6} | {:6} | {:6} | {:6} |".format(ntf,
nf, nm, ne, nte))
f.close()
```

Son exécution affiche l'histogramme de qualité de l'air pour l'année 2017 pour la ville de Saint-Denis :

Pollution				
Très Faible	Faible	Moyenne	Élevée	Très élevée
10	262	83	9	0

Les trois activités **10** à **12** illustrent la richesse des traitements de fichier pouvant être programmés à partir d'un seul modèle d'algorithme de lecture ligne à ligne du fichier et de traitement successif de ses lignes. Ces activités peuvent être transposées simplement à d'autres jeux de données selon le choix du domaine d'application ciblé. Tous les traitements consistant en des comptages, des détections de seuils ou des filtrages de données peuvent être programmés de cette manière.

Les traitements plus complexes sont ceux où toute l'information doit être chargée simultanément en mémoire avant de commencer le traitement, comme dans les algorithmes de tris ou de croisement entre données.

Analyse et débats

La découverte de la richesse des données collectées et disponibles et de la puissance des traitements possibles amène inéluctablement à se poser la question de leur usage. Les débats sur l'usage des traitements massifs de données peuvent être menées de manière plus éclairée après avoir découvert ce que l'on peut en programmer.

Enjeu 1. L'ouverture des données publiques

L'ouverture des données publiques est un phénomène relativement récent qui ouvre des possibilités importantes à condition que les utilisateurs potentiels aient les capacités techniques de s'en saisir et les moyens d'exploiter et de « faire parler » les données.

Enjeu 2. L'exploitation du *« big data »*, science ou surveillance ?

Le « *big data* », par la quantité des données recueillies, offre des potentialités pouvant être considérées comme des chances ou comme des menaces. La puissance des algorithmes, répétant des millions de fois des calculs élémentaires permet d'envisager des traitements auparavant impossibles à réaliser. La question de l'éthique peut être posée avec les élèves, en particulier la question des limites que la société choisit ou non de poser aux possibilités de la science.

Enjeu 3. Les données personnelles : protection et usages

Les données à caractère personnel sont protégées par la loi. C'est d'ailleurs la seule manière de protéger ces données, car une fois numérisées, les données peuvent techniquement être copiées sans limite et croisées entre elles, dévoilant par recoupement de nouvelles informations potentiellement à caractère personnel non divulguées précédemment. L'enjeu de protection est donc essentiellement juridique.

Enjeu 4. Les *data centers* et leurs impacts sur l'environnement

L'impact sur l'environnement du stockage et des traitements massifs de données augmente rapidement alors que le réchauffement climatique s'accélère. Sans tenir un discours excessivement alarmiste, il est important de poser cette contradiction. La dématérialisation d'activités, censée apporter des économies de matière et d'énergie, occasionne en retour une augmentation de la consommation des réseaux et des centres de données.

Pour aller plus loin

Les activités complémentaires proposées abordent des algorithmes de traitement de données nécessitant le stockage en mémoire des données et l'utilisation de méthodes de programmation au programme de la spécialité Numérique et Sciences informatiques (NSI) en classe de première. Ces activités ont donc seulement un objectif de découverte et sont à mener en mode débranché, sans chercher à programmer effectivement les algorithmes abordés.

Activité 13. Trier les données

- **Activité débranchée.**
- **Durée estimée : 30 minutes.**

La question de la découverte d'un algorithme systématique de tri diffère sensiblement de l'activité manuelle consistant à trier un paquet de cartes. En effet, l'humain dispose d'une capacité de vision globale lui permettant d'avoir une planification très

élaborée de son action mêlant des actions locales pour placer une carte par rapport à une autre, et des actions globales positionnant plutôt les cartes les plus hautes à tel endroit et les plus basses à un autre.
Anticiper un algorithme pouvant être exécuté par une machine oblige à se limiter à des actions élémentaires stéréotypées. On a choisi ici de permettre un échange de carte, et de tester deux cartes pour savoir si elles sont dans le bon ordre.
Les cartes sont supposées être rangées dans un tableau porte-cartes qui simule la structure de la mémoire et fournit des numéros de cases permettant de désigner les cases dont on va observer les cartes.
Un algorithme possible avec les instructions proposées est l'algorithme du tri à bulle.

```
pour i de 1 à 19
    pour j de 1 à 19
         si cartes à la position j et à la position j+1 ne sont
         pas dans l'ordre
            échanger la carte à la position j avec la carte à la
            position j+1
```

L'intérêt de donner l'algorithme écrit par un élève à un autre élève est de le faire exécuter sans *a priori*, machinalement, sans connaître l'intention du rédacteur.
On peut ainsi plus facilement constater les erreurs ou opérations inutiles prévues par le rédacteur de l'algorithme, et engager alors un dialogue pour améliorer l'algorithme proposé.

Activité 14. Traiter plusieurs tables pour croiser des données

- Activité **débranchée.**
- **Durée estimée : 30 minutes.**

Cette activité présente un problème de croisement de deux fichiers dont la difficulté vient de la recherche de l'ordre dans lequel parcourir les deux fichiers pour obtenir les informations demandées.
Le fait de disposer seulement des fichiers imprimés empêche de pouvoir les trier en mémoire et oblige à les consulter dans l'ordre où ils sont imprimés.

1. Il suffit de rechercher ligne à ligne dans le premier fichier la ligne comportant le nom du conducteur recherché, puis de noter le lieu de départ, le lieu d'arrivée et la date. Ensuite, il faut rechercher dans le second fichier ligne à ligne les passagers demandant un covoiturage ce jour-là, avec le même départ et la même arrivée.
L'ordre de parcours est simple, d'abord le premier fichier, puis le second.

2. L'organisateur du covoiturage doit repérer toutes les combinaisons possibles ; pour cela, il doit parcourir plusieurs fois chaque fichier pour trouver pour chaque conducteur tous les passagers qu'il pourrait emmener, ou pour chaque passager, tous les conducteurs qui pourraient l'emmener. Il y a plusieurs algorithmes possibles : tous nécessitent de parcourir un des fichiers une fois du début à la fin et l'autre autant de fois qu'il y a de fiches dans le premier fichier.

3. La troisième question est en fait comparable à la première. Il faut commencer par rechercher la fiche du passager concerné, puis parcourir le fichier des conducteurs. En l'absence de tri possible par date de départ, il faut de toute façon scruter tout le fichier des conducteurs à la recherche de ceux qui effectuent le même trajet le jour même, la veille ou le lendemain.