



Chapitre 4

Manipulation de fichiers

TP

Simon Dauguet
simon.dauguet@gmail.com

20 novembre 2025

1 Piles

2 Fichiers

Exercice 1 :

On considère un fichier `Diophante1.txt`, qui contient un texte en minuscules et sans accent. Faire une procédure `Dio()` -> **None** qui lise chaque ligne, compte le nombre de lettre 'e' puis et ajoute dans le fichier `Diophante2.txt` la ligne suivie d'un espace, du nombre de 'e' de la ligne et d'un 'e'. Ce qui donnerait

souvent, pour s'amuser, les hommes d'equipage prennent des albatros, vastes oiseaux des mers, qui suivent, indolents compagnons de voyage, ...

souvent, pour s'amuser, les hommes d'equipage 6e prennent des albatros, vastes oiseaux des mers, 7e qui suivent, indolents compagnons de voyage, 4e ...
--

En supposant que l'on définisse `DiophanteN.txt` défini par récurrence comme le résultat de la lecture du fichier `livreN-1.txt`, existe-t-il un rang à partir duquel le fichier n'évolue plus ?

Remarque :

On rappelle que le système de fichiers utilisé par Capytale est un système virtuel. Donc vous ne pouvez pas voir les fichiers créés (seulement les fichiers extérieurs importés au lancement). La commande `os.listdir()` vous permet de voir la liste des fichiers et dossiers du dossier de travail courant.

Exercice 2 :

On souhaite créer un exercice de calcul qui génère deux nombres aléatoires entre 2 et 15, puis demande le résultat de la multiplication. Proposer une procédure `exerciceur()` -> **None** qui demande le nom de l'utilisateur (qui deviendra le nom du fichier) puis demande le résultat de 10 multiplications. Dès qu'il y a une erreur (ou à la fin) le nombre de sans faute est écrit dans le fichier, puis le produit qui a posé problème.

Par exemple, si la personne se nomme **Maya**, fait un sans faute sur les six premières questions, une erreur au calcul 12×7 , puis aucune erreur, un fichier nommé `Maya.txt` sera créé ainsi

6 sans faute erreur : 12*7 3 sans faute

Exercice 3 (Mot caché) :

Dans le fichier `mot_cache.txt`, il y a une longue suite de caractère non alpha-numérique. Quelque part, dans ce fouillis, ce cache quelques lettres en minuscules. Lire le fichier en question et récupérer les lettres qui s'y trouve pour former le mot caché.

Exercice 4 (Produit mystère) :

Dans le fichier `Nombre.txt`, un très grand nombre est inscrit. Les quatre chiffres consécutifs dont le produit est le plus grand est $9 \times 9 \times 8 \times 9 = 5832$. Trouver les 13 chiffres consécutifs dont le produit est maximal (en renverra le produit, la séquence des 13 chiffres et la position du premier).

Exercice 5 (Lettres bien encadrées) :

Dans le fichier `Lettres.txt`, trouver les lettres minuscules qui sont entourées d'exactly 3 majuscules (à droite et à gauche). Les lettres minuscules forment un mot mystère.

Exercice 6 (Entier premier de Gauss) :

On considère l'entier écrit dans le fichier `gauss.txt`. Essayer de l'écrire en plusieurs lignes de différentes longueurs. Que peut-on remarquer si on prend du recul ?

Attention, vous êtes ici maître à bord. Aucune indication n'est fournie (*a priori*) et vous devez donc vous organisez seul pour répondre au problème. À vous de prendre des initiatives.

Remarque : Un entier premier de Gauss est entier premier qui ne peut s'écrire sous la forme de la somme de deux carrés. On pourrait essayer de montrer que l'entier manipulé ici est un entier de Gauss.

Exercice 7 (Jouons dans les arbres) :

Le but de cet exercice est simplement de s'habituer à la manipulation de fichiers et dossiers en ligne de commandes (sans souris). Tout se passe donc dans l'interpréteur et pas dans l'éditeur pour cet exercice.

1. Créer un dossier `Doss1/Doss2/Doss3`.
2. Déplacer vous dans le dossier `Doss3`.
3. Sans vous déplacer, créer un fichier `Doss1/Doss2bis/fichier.txt` qui contiendra un mot (celui que vous voulez).
4. Déplacer pour revenir à la racine. Puis faire la liste de ce que contient le dossier `Doss1`.
5. Faire la liste de ce que contient le dossier `Doss2bis`.
6. En restant dans la racine, créer un dossier `Doss3` dans le dossier `Doss1`.
7. Renommer le dossier `Doss3` en `Doss2ter`.
8. Supprimer le dossier `Doss1`.

Exercice 8 (Jeu de piste) :

Cet exercice se fait à deux. Commencer donc par choisir un binôme. Chacun crée un nouveau fichier sur Capytale que vous appellerez `TP4_NextNothing_Prenom1_Prenom2`. Puis, dans ce fichier, vous ferez :

1. Vous ferez une fonction `MakeNothing()` -> `int` qui va créer un dossier `Nothing`. La fonction devra choisir un nombre aléatoire $N \in \llbracket 500, 1000 \rrbracket$ correspondant au nombre de fichiers que vous aller créer. Vous créez ensuite N fichiers dans le dossier `Nothing` dont le nom sera un numéro aléatoire dans $\llbracket 10000, 100000 \rrbracket$. Chaque fichier contiendra le texte `"The next nothing is {num}"` où `{num}` sera le numéro prochain fichiers créé. Dans le dernier fichier créé, vous mettrez une phrase mystère que votre adversaire devra trouver. La fonction devra renvoyer le numéro du premier fichier créé (le point de départ du jeu de piste). Évidemment, on s'assurera de ne pas utiliser deux fois le même numéro de fichier.

2. Vous partagerez ensuite votre fichier avec votre binôme.
3. Une fois dans le fichier de votre adversaire, commencez par ne pas tricher et réduire le code **MakeNothing** qui sera présent pour ne pas voir la phrase mystère choisie.
4. Compiler et lancer la fonction **MakeNothing()**.
5. Puis, remonter le jeu de pistes. Autrement dit : faire une fonction **Nothing(N:int)** -> **tuple** qui, à partir de l'entier renvoyer par la fonction **MakeNothing**, ira lire chacun des fichier du dossier **Nothing**, prendra le nombre qui y apparaîtra (tant qu'il y en a) et renverra le numéro du dernier fichier et la phrase mystère contenue dans le dernier fichier au bout de la chaîne.

Vous noterez que vous pouvez jouer depuis chez vous : Capytale est en ligne, donc nulle besoin d'être en séance de TP pour pouvoir jouer.