

TP de Langages de script n° 3 bis : Exercices supplémentaires

Exercice 1 : crible d'Ératosthène

1. Écrire une fonction `est_premier` qui teste si un entier est premier en regardant s'il possède un diviseur parmi les entiers inférieurs à sa racine carrée (fonction `sqrt` du module `math`).
2. Écrire une fonction retournant la liste des nombres premiers inférieurs à son paramètre.
3. Une méthode plus efficace pour calculer cette liste est la méthode dite du *crible d'Ératosthène*: elle consiste à écrire la liste des entiers de 2 à la valeur maximale souhaitée, puis à répéter l'opération suivante: marquer la plus petite valeur qui n'a pas encore été rayée ni marquée, puis rayer ses multiples. Programmer cet algorithme.
4. Dans les deux méthodes, est-il nécessaire de faire varier i de 1 jusqu'à n ?

Exercice 2 : approximation de $\sin x$

On souhaite calculer une valeur approchée de $\sin x$ à l'aide son développement en série de Taylor:

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + (-1)^{2n+1} \frac{x^{2n+1}}{(2n+1)!}$$

Écrire une fonction prenant deux paramètres, x et ε , et calculant une valeur approchée de $\sin x$ avec une erreur inférieure à ε .

Exercice 3 : recherche de fichiers

Dans cet exercice, vous allez programmer un script `trouver.py` semblable à la commande Unix `find`, qui permet de rechercher des fichiers dans une arborescence selon de multiples critères. Il faudra veiller à générer une page d'aide contenant une description succincte de chaque option.

Attention : sauf exception, toutes les options sont cumulatives.

1. Créer une première version du script permettant d'afficher la liste exhaustive et récursive des fichiers *et répertoires* contenus dans un répertoire passé en argument. Si l'argument est absent, on considérera le répertoire courant.
2. Ajouter une option `-e` ou `--empty` au script: en présence de cette option, seuls les fichiers de taille 0 et les répertoires vides seront listés.
3. Ajouter une option `-s` ou `--show` au script. Cette option doit être suivie d'un argument valant `visible`, `hidden` ou `all`. En présence de cette option, le script doit respectivement afficher uniquement les fichiers visibles, uniquement les fichiers cachés, ou tous les fichiers. Le comportement par défaut est d'afficher uniquement les fichiers visibles.
4. Ajouter une option `--b` ou `--big` suivie d'un argument entier; en présence de cette option, le script ne doit lister que les fichiers ayant une taille supérieure à la taille indiquée (en octets).
5. Ajouter une option `-n` ou `--name` suivie d'une chaîne de caractères représentant une expression régulière. Le script ne doit afficher que les fichiers ou répertoires dont le nom correspond à cette expression régulière.

6. Ajouter une option `-d` ou `--depth` au script; cette option doit être suivie d'un argument n entier. En présence de cette option, votre script doit afficher uniquement les fichiers ou répertoires à une profondeur inférieure à n depuis le répertoire de départ.

Choisir une extension (au moins) parmi les questions suivantes:

7. Ajouter une option `-o` ou `--old` au script suivie d'un argument entier n , en présence de laquelle le script ne doit lister que les fichiers dont la date de dernier accès est antérieure d'au moins n jours à la date courante.
8. Ajouter une option `--clean`, en présence de laquelle le script doit supprimer les fichiers et répertoires vides (on pourra utiliser les fonctions `remove` et `removedirs` du module `os`).
9. Ajouter une option `--gzip`, en présence de laquelle le script devra compresser les fichiers trouvés (utiliser la fonction `system` du module `os`, et la commande `gzip`).
10. Ajouter une option `-r` ou `--rights` suivie d'un argument entier en octal (commençant par un 0) ou en binaire (commençant par 0b) représentant des permissions. En présence de cette option, le script ne doit afficher que les fichiers ou répertoires possédant aux moins des droits équivalents. Modifier le script pour prendre en compte les droits de traversée des répertoires si nécessaire (*difficile*).