Exercices Python

Instructions de base

Lire un nombre, écrire s'il est positif, négatif ou nul. *(en dur, puis en procédure, puis en fonction)*
Lire un nombre, écrire s'il est pair ou impair.

Fonction et procédure
De deux nombres, afficher/retourner le plus petit des deux.
De deux nombres, afficher/retourner le plus grand des deux.
De trois nombres, afficher/retourner le plus petit et le plus grand.

Lire un jour de semaine, si c'est lundi écrire « c'est reparti !», si c'est mercredi écrire « jour des enfants », si c'est vendredi écrire « bientôt le week-end » si c'est samedi ou dimanche écrire « bon week-end » sinon écrire « bonne semaine »

Lire une couleur, si c'est rouge magenta, bleu cyan ou jaune, écrire « couleur primaire ». Si c'est orange, violet ou vert, écrire « couleur secondaire ». Sinon écrire « autre couleur » .

Calculer la somme des n premiers nombres (itératif et récursif)
Calculer la somme des nombres pairs inférieurs ou égaux à N.

Algorithmique itérative mathématiques

→ Déterminer le plus grand n tel que la somme des n premiers nombres est inférieure à 15000.
→ Calculer factorielle.
→ faire une fonction qui calcule la puissance.
→ passer d'un nombre en base 10 à base 2 et inversement.
→ suite de Fibonacci (« Un homme met un couple de lapins dans un lieu isolé de tous les côtés par un mur. Combien de couples obtient-on en un an si chaque couple engendre tous les mois un nouveau couple à compter du troisième mois de son existence ? »)
→ Triangle de pascal (sur chaque ligne n+1 on retrouve la somme des nombres de la ligne n, deux à deux, la représentation spatiale de ce triangle possède de nombreuses propriétés mathématiques étonnantes, dignes des meilleures théories du complot)
→ Suite de Syracuse qui converge sur 1 4 2 1 4 2 1 4 2 etc.

Les couples

Un individu est défini par un couple contenant son nom et son age ex ("Bernard", 60)
Créer une méthode qui permet de retourner le nom d'un individu getNom(individu):
Créer une méthode qui permet de retourner l'age d'un individu getAge(individu):
Créer une méthode qui permet d'afficher un individu ex : Bernard, 60 ans.
Créer une méthode qui permet de comparer deux individus selon l'âge puis le nom pour indiquer si un individu est « plus petit » qu'un autre.

Les listes indexées (récursif et itératif).

Créer trois listes d'individus
LISTE_INDIVIDUS=[("Bernard", 60),("Céline", 40),("Jade", 30),("Olivia", 15),("Kevin", 20)]
LISTE_INDIVIDU_SEUL=[("Kevin", 20)] ; LISTE_VIDE=[]

Afficher une liste
Inverser une liste
Vérifier si une liste est un palindrome
Vérifier si un nom de personne apparaît au moins une fois dans la liste
Supprimer les personnes qui ont le même âge
Inverser ordre de la liste
Trier une liste :
tri bulle, tri insertion (on prend le premier, on l'insère trié), tri sélection (on met le plus petit en tête), tri fusion, tri rapide (avec un pivot, on trie les deux sous-listes et on concatène avec le pivot au milieu), tri par tas () …

Dictionnaires.

Créer un répertoire de personnes plutôt qu'un couple nom/age, on fait un dictionnaire nom->age
Écrire trois procédures :
→ afficher toutes les clés → afficher toutes les valeurs → afficher tous les couples "clé/valeur".
Écrire une fonction qui indique si deux personnes ont le même âge.
Retourner le dictionnaire inverse : age -> liste de personnes
Copier-coller un texte quelconque sur le net et retourner un dictionnaire qui retourne la fréquence de chaque mot.

Objet.

Jeu du Memory
Considérer un paquet de 2*n cartes, organisées par paires, qui seront présentées, faces cachées, à un joueur. Ce joueur doit retrouver les paires en retournant deux cartes, l'une après l'autre. Lorsqu'il constitue une paire, elle est retirée. Le programme doit compter le nombre de coups effectués pour retirer toutes les paires et permettre de choisir le nombre de cartes à générer.
Une **carte** possède deux attributs : une valeur entière et une lettre (minuscule ou majuscule). Toutes les valeurs sont différentes, toutes les lettres sont différentes.
Les cartes sont mélangées avant d'être disposées sur une **grille** à peu près carrée où chaque case est représentée par sa position (par exemple un index ou un couple (x,y) ou un couple (lettre, nombre) comme aux échecs).
Évolution : faire évoluer votre code pour permettre de choisir le nombre de cartes identiques à retrouver, plutôt que ce nombre soit égal à 2.

Jeu du plus ou moins
Nous allons considérer une autre sorte de paquet de cartes contenant une valeur comme attribut et une couleur. Créer une classe commune à ces deux paquets de cartes qui va contenir toutes les informations et fonctions liées à la gestion de la valeur de la carte. Modifier votre classe Memory pour créer une classe CarteMemory qui ajoute une lettre et les fonctionnalités liés aux cartes du memory. Vérifier que votre programme précédent fonctionne toujours très bien.
Créer une classe Carte32 qui ajoute une couleur à une Carte et permet de gérer les cartes d'un jeu de 32 cartes (valeurs de 7 à 14, couleur rouge ou noir). Proposez un jeu qui permet de deviner, sur un tirage aléatoire du paquet de 32 cartes, si la carte suivante sera plus grande ou plus petite que la précédente et qui enregistre la plus grande suite de bonnes prédictions réalisée sans erreur sur le paquet de 32 cartes.

Exceptions.

Reprendre les menus proposés avec une liste et gérer les mauvaises saisies sur ces menus en utilisant l'exception sur le mauvais type de données saisi et en levant notre propre exception ChoixInvalideException.

Modules.

Découvrir le module tKinter pour créer une fenêtre interactive avec l'utilisateur qui lui demande son prénom et affiche dans une autre fenêtre le nombre de lettres de ce nom.

Découvrir le module Vispy et dessiner une maison simple en 3D (parallélépipède rectangle surmonté d'une pyramide.
http://vispy.org/vispy.html#module-vispy
https://deptinfo-ensip.univ-poitiers.fr/ENS/doku/doku.php/stu:python_gui:vispy

Sérialisation.

Proposer une sauvegarde et un chargement d'une partie de Memory en cours.