Travaux pratiques - Modularité - Correction

Exercice 1 Utilisation des modules Random, OS et Datetime

1. **Module `random`** : Écrivez une fonction `lancer_des(n)` qui simule le lancer d'un dé à n faces. La fonction doit renvoyer le résultat du lancer.
2. **Module `os`** : Écrivez une fonction `liste_fichiers(repertoire)` qui prend en paramètre un chemin de répertoire et renvoie la liste des fichiers présents dans ce répertoire.
3. **Module `datetime`** : Écrivez une fonction `afficher_date_actuelle()` qui affiche la date et l'heure actuelles.
4. Programme principal : Dans le programme principal, appelez chacune des fonctions que vous avez écrites pour tester leur fonctionnement.
   Affichez les résultats de manière claire et lisible.

```
import random
import os
import datetime

def lancer_des():
    return random.randint(1,6)

def liste_fichiers(repertoire):
    return os.listdir(repertoire)

def afficher_date_actuelle():
    print(datetime.datetime.now())

print(lancer_des())
print(liste_fichiers('/home/mikael/athle'))

afficher_date_actuelle()
```

Exercice 2 Implémenter en python le module de probabilités du TD et vérifier les résultats attendus de l'exemple. **Voir correction TD**

Exercice 3 Implémenter en python le module de gestion des contacts du devoir maison.

Créer le fichier `contacts.py`, puis le programme principal de test dans le fichier `test_contact.py` . **Voir correction DM**

Exercice 4

PARTIE 1 - Création d'un module de géodésie

Le but de ce TP est de créer un module Python pour effectuer des calculs géodésiques. Votre module `geodesie.py` devra implémenter les fonctionnalités suivantes :

- le calcul de la longueur d'un parallèle
- la distance entre deux points sur un même parallèle
- la distance entre deux points sur un même méridien
- la distance entre deux points quelconques sur la sphère terrestre
- une méthode pour calculer la longueur d'un mille nautique.

Formules mathématiques

Dans la suite de ce TP, nous noterons R le rayon de la Terre, avec $R = 6371$ km.

1. **Longueur d'un parallèle** : La longueur d'un parallèle à une latitude Phi (en degrés) est donnée par la formule :
$$L = 2\pi \times R \times \cos \Phi$$
2. **Distance entre deux points sur le même parallèle** : La distance entre deux points sur le même parallèle (en degrés) est donnée par la formule :
$$D = R \times \Delta\text{longitude} \times \cos \Phi$$
où Δlongitude est la différence de longitude entre les deux points, exprimée en radian.
3. **Distance entre deux points sur le même méridien** : La distance entre deux points sur le même méridien (en degrés) est donnée par la formule :
$$D = R \times \Delta\text{Latitude}$$
où ΔLatitude est la différence de latitude entre les deux points, exprimée en radian.
4. **Distance entre deux points quelconques sur la sphère terrestre** : La distance entre deux points quelconques de latitudes Φ_1 et Φ_2 sur la sphère terrestre (en degrés) est donnée par la formule :
$$D = R \times \arccos\left(\sin \Phi_1 \times \sin \Phi_2\right) + \cos \Phi_1 \times \cos \Phi_2 \times \cos(\Delta\text{Longitude}))$$
5. **Longueur d'un mille nautique** : La longueur d'un mille nautique est définie comme une minute d'arc (1/60 de degré), ce qui équivaut à environ 1.852 kilomètres. Il s'agit de la distance parcourue sur un même méridien si l'on se déplace d'1/60 de degré sur ce méridien.

- Vous testerez votre module à l'aide d'un programme de test `test_geodesie.py`
- Attention, les fonctions $sinus$ et $cosinus$ utilisent des degrés exprimés en radian. Il faut donc convertir les latitudes et longitudes en radian. Le module `math` propose une fonction pour cela. Reportez-vous à l'aide en ligne de Pyhton pour plus d'information.
- Les longitudes à l'ouest doivent être notées négativement. Exemple : 2,7° W doit être traduit par -2.7°.
- Les latitudes au sud doivent être notées négativement. Exemple : 34,6° S doit être traduit par -34.6°.

```python
# geodesie.py
import math

def longueur_parallele(rayon_terre, latitude):
    """
    Calcule la longueur d'un parallele a une latitude donnee.

    :param rayon_terre: Rayon de la Terre en kilometres.
    :param latitude: Latitude en degres.
    :return: Longueur du parallele en kilometres.
    """
    phi = math.radians(latitude)
    return 2 * math.pi * rayon_terre * math.cos(phi)

def distance_parallele(rayon_terre, latitude, long1, long2):
    """
    Calcule la distance entre deux points sur le meme parallele.

    :param rayon_terre: Rayon de la Terre en kilometres.
    :param latitude: Latitude en degres.
    :param long1: Longitude du premier point en degres.
    :param long2: Longitude du deuxieme point en degres.
    :return: Distance entre les deux points en kilometres.
    """
    phi = math.radians(latitude)
    delta_longitude = math.radians(abs(long1 - long2))
    return rayon_terre * delta_longitude * math.cos(phi)

def distance_meridien(rayon_terre, lat1, lat2, long):
    """
    Calcule la distance entre deux points sur le meme meridien.

    :param rayon_terre: Rayon de la Terre en kilometres.
    :param lat1: Latitude du premier point en degres.
    :param lat2: Latitude du deuxieme point en degrs.
    :param long: Longitude commune en degres.
    :return: Distance entre les deux points en kilometres.
    """
    delta_latitude = math.radians(abs(lat1 - lat2))
    return rayon_terre * delta_latitude
```

```
def distance_points(rayon_terre, lat1, long1, lat2, long2):
    """
    Calcule la distance entre deux points quelconques sur la sphere terrestre.

    :param rayon_terre: Rayon de la Terre en kilometres.
    :param lat1: Latitude du premier point en degres.
    :param long1: Longitude du premier point en degres.
    :param lat2: Latitude du deuxieme point en degres.
    :param long2: Longitude du deuxieme point en degres.
    :return: Distance entre les deux points en kilometres.
    """
    phi1, phi2 = math.radians(lat1), math.radians(lat2)
    delta_longitude = math.radians(long2 - long1)
    return rayon_terre * math.acos(math.sin(phi1) * math.sin(phi2) + math.cos(phi1)
        * math.cos(phi2) * math.cos(delta_longitude))

def longueur_mille_nautique(rayon):
    """
    Calcule la longueur d'un mille nautique sur la sphere terrestre.

    :param rayon: Rayon de la sphere terrestre.
    :return: Longueur d'un mille nautique en unites de longueur terrestre.
    """
    # Un mille nautique est defini comme une minute d'arc sur un meridien
    # 1 degre = 60 minutes d'arc
    # Formule : longueur = (1 / 60) * 2 * pi * rayon
    return (1 / 60) * 2 * math.pi * rayon
```

Cas pratiques

1. Compléter le tableau suivant :

Latitude	Equateur	Tropique	35.6°	40.7°	48.5°	49°	90°
Longueur du parallèle	40030	36737	32548	30348	26524	26262	0

2. Compléter le tableau des coordonnées ci-dessous puis modifier le programme principal pour compléter le tableau des distances.

Tableau des coordonnées sur la sphère terrestre.

	St Brieuc	Paris	New York	Tokyo	Melbourne
Latitude	48.5	48.8	40.7	35.7	-37.8
Longitude	-2.7	2.3	-74	139.7	144.9

Tableau des distances

Distances en km	St Brieuc	Paris	New York	Tokyo	Melbourne
St Brieuc	0	368	5504	8923	17157
Paris	368	0	5836	9718	16790
New York	5504	5836	0	10848	16677
Tokyo	8923	9718	10848	0	8190
Melbourne	17157	16790	16677	8190	0