Ce document a été mis en ligne par l'organisme FormaV®.

Toute reproduction, représentation ou diffusion, même partielle, sans autorisation préalable, est strictement interdite.

Pour consulter tous nos sujets d'annales et corrigés du Bac Général Maths + NSI, veuillez visiter :
bac-maths-nsi.fr/annales-sujets-et-corriges

Centres étrangers - mai 2022 - sujet 1 (corrigé)

Exercice 1 (Dictionnaires et listes)

1. (a) On obtient lundi avec l'instruction `jours[1]`.
   (b) `jours[18%7]` correspond à `jours[4]`, c'est-à-dire à jeudi.
2. Il faut saisir l'instruction : `numero_jour =(jours.index(j) + n )% 7`
3. (a) Pour obtenir le nombre de jours au mois de mars, on doit écrire : `mois[3][1]`
   (b) Il faut saisir l'instruction : `mois[(numero_mois + x) % 12][0]`
4. (a) L'instruction `mois[date[2]][1]` renvoie `31`.
   (b) La fonction suivante convient :

```
def jour_suivant(date):
    nom_j = date[0]
    j = date[1]
    m = date[2]
    a = date[3]
    nom_jour_plus = jours[(jours.index(nom_j) + 1 )% 7]
    nbe_jour_mois = mois[date[2]][1]
    jour_plus = (j + 1) % nbe_jour_mois
    if nbe_jour_mois == j:
        mois_plus = (m + 1)%12
    else :
        mois_plus = m
    if m == 12 and j == 31:
        annee_plus = a + 1
    else :
        annee_plus = a
    return (nom_jour_plus, jour_plus, mois_plus, annee_plus)
```

Exercice 2 (Files et POO)

1. Le code suivant convient :

```
panier1.enfiler((31002, "café noir", 1.50, 50525))
```

2. Le code suivant convient :

```
def remplir(self, panier_temp):
    while not panier_temp.est_vide() :
        article = panier_temp.defiler()
        self.enfiler(article)
```

3. Le code suivant convient :

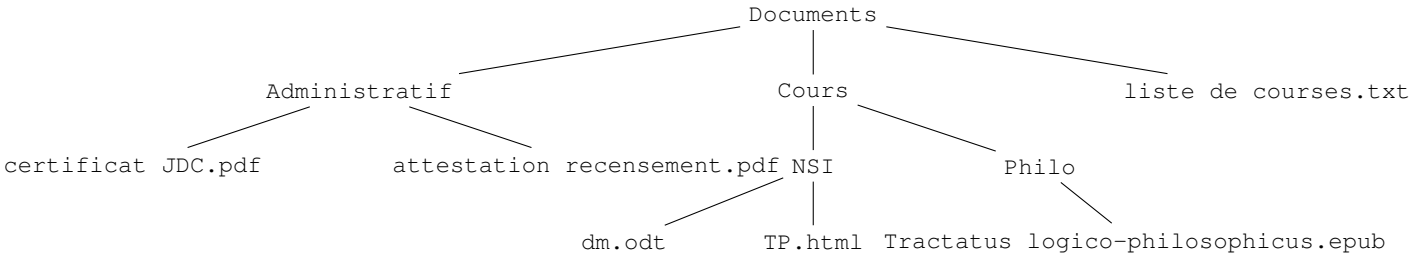
```
def prix_total(self):
    p_temp = Panier()
    montant = 0
    while not self.est_vide() :
        article = self.defiler()
        montant = montant + article[2]
        p_temp.enfiler(article)
    while not p_temp.est_vide() :
        article = p_temp.defiler()
        self.enfiler(article)
    return montant
```

4. Le code suivant convient :

```
def horaire_scan(self):
    if self.est_vide():
        return 0
    premier_article = self.defiler()[3]
    dernier_article = premier_article
    while not self.est_vide() :
        dernier_article = self.defiler()[3]
    return dernier_article - premier_article
```

Exercice 3 (Dictionnaires)

1. On a l'arbre suivant :



2. (a) La fonction suivante convient :

```
def parcourir(racine, adr):
    dossier = racine
    for nom_dossier in adr:
        dossier = dossier[nom_dossier]
    return dossier
```

   (b) L'instruction `afficher(Documents, ["Cours", "NSI"], "TP.html")` permet d'afficher la taille en Ko du fichier `TP.html`, c'est-à-dire `60`.

3. (a) L'erreur est à la ligne 3.
   Il faut écrire `dossier[nom_fichier] = taille` à la place de `taille = dossier[nom_fichier]`.

   (b) La fonction suivante convient :

```
def ajouter_dossier(racine, adr, nom_dossier):
    dossier = parcourir(racine, adr)
    dossier[nom_dossier]={}
```

4. La fonction suivante convient :

```
def taille(dossier):
    s = 0
    for k in dossier.keys():
        s = s + dossier[k]
    return s
```

Exercice 4 (Bases de données et SQL)

1. (a) L'attribut `id_mesure` peut jouer le rôle de clé primaire. En effet, chaque valeur de cet attribut est unique (ce qui n'est pas le cas des autres attributs de cette relation, par exemple, il est possible d'avoir plusieurs fois la même température).

   (b) Les attributs `Mesures.id_centre` et `Centres.id_centre` vont permettre de faire une jointure entre la relation `Centres` et la relation `Mesures` (`Mesures.id_centre` est une clé étrangère).

2. (a) Cette requête permet d'afficher tous les centres qui ont une altitude supérieure à 500 m. Plus exactement, on affichera l'`id_centre`, le nom, la latitude, la longitude et l'altitude de tous les centres qui ont une altitude supérieure à 500 m.

   (b) La requête suivante convient :

```
SELECT nom_ville
FROM Centres
WHERE altitude > 700 AND altitude < 1200
```

   (c) La requête suivante convient :

```
SELECT longitude, nom_ville
FROM Centres
WHERE longitude > 5 ORDER BY nom_ville
```

3. (a) Cette requête permet d'afficher l'ensemble des données (`id_mesure`, `id_centre`, `date`, `température`, `pression` et `pluviométrie`) des mesures effectuées le 30 octobre 2021.

   (b) La requête suivante convient :

```
INSERT INTO Mesures VALUES (3650, 138, 2021-11-8, 11, 1013, 0)
```

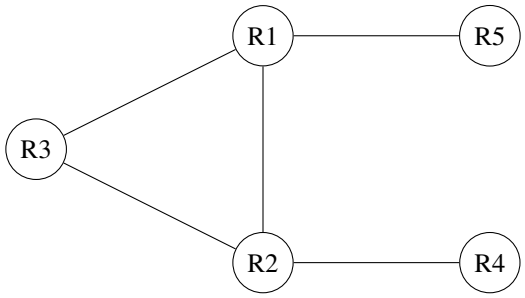
4. (a) Cette requête permet d'afficher l'ensemble des données (`id_centre`, `nom`, `latitude`, `longitude` et `altitude`) du centre qui a la latitude minimum (le centre situé le plus au sud).

   (b) La requête suivante convient :

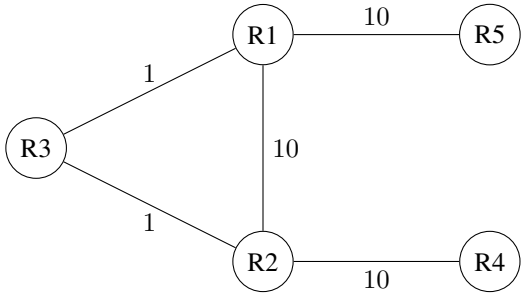
```
SELECT DISTINCT nom_ville
FROM Centres
JOIN Mesures ON Mesures.id_centre = Centres.id_centre
WHERE temperature < 10 AND date > 2021-09-30 AND date < 2021-11-1
```

Exercice 5 (Architecture matérielle, réseaux et protocoles)

1. (a) Dans un SoC (System on Chip) les différents composants (CPU, RAM, GPU, circuit Wifi…) sont tous intégrés sur la même puce (le SoC). Dans un ordinateur « classique » tous ces composants sont séparés.
   (b) Seul le Broadcom BCM271 possède une interface réseau de type Ethernet, donc, seul ce SoC peut être connecté à un réseau filaire.
   (c) On peut comparer le nombre de coeurs (4 contre 1) et la fréquence de base (1,5 GHz contre 700 MHz). Le Broadcom BCM271 est donc plus « puissant » que le Broadcom BCM2835.
2. (a) L'adresse MAC est liée au matériel : chaque carte réseau (Ethernet ou Wifi) possède sa propre adresse MAC (il n'existe pas dans le monde, deux cartes réseau (Ethernet ou Wifi) qui possèdent la même adresse MAC). On retrouve l'adresse MAC au niveau de la couche accès réseau du modèle TCP/IP, elle permet d'identifier les machines sur un réseau (au niveau de la couche accès réseau).
   (b) `10.0.2.15` correspond à une adresse IP (qui permet d'identifier les machines au niveau de la couche IP du modèle TCP/IP).
   (c) L'adresse `10.0.2.2` correspond à une passerelle (gateway).
3. (a) On a le schéma suivant :



   (b) On a la route suivante : R4 – R2 – R1 – R5
4. En ajoutant les coûts sur les liaisons, on a le schéma suivant :



On en déduit que le chemin qui doit minimiser le coût total est la route : R4 – R2 – R3 – R1 – R5 (coût = 22).

Copyright © 2027 FormaV®. Tous droits réservés.

FormaV® est une marque déposée auprès de l'INPI sous le numéro 5227897.

Ce document a été élaboré par FormaV® avec le plus grand soin afin d'accompagner chaque apprenant vers la réussite de ses examens. Son contenu (textes, graphiques, méthodologies, tableaux, exercices, concepts, mises en forme) constitue une œuvre protégée par le droit d'auteur.

Toute copie, partage, reproduction, diffusion ou mise à disposition, même partielle, gratuite ou payante, est strictement interdite sans accord préalable et écrit de FormaV®, conformément aux articles L.111-1 et suivants du Code de la propriété intellectuelle. Dans une logique anti-plagiat, FormaV® se réserve le droit de vérifier toute utilisation illicite, y compris sur les plateformes en ligne ou sites tiers.

En utilisant ce document, vous vous engagez à respecter ces règles et à préserver l'intégrité du travail fourni. La consultation de ce document est strictement personnelle.

Merci de respecter le travail accompli afin de permettre la création continue de ressources pédagogiques fiables et accessibles.