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

# Amérique du sud - septembre 2024 - sujet 2 (corrigé)

**Exercice 1 (Récursivité et langage SQL)**

1. La fonction `resoudre` est une fonction récursive car elle s'appelle elle-même (ligne 28).
2. On a les codes suivants :
   - Ligne 9 : `if cible in valeurs:`
   - Ligne 14 : `if len(valeurs) <= 1:`
3. Les fonctions suivantes conviennent :

```
def add(a,b):
    return a+b

def mul (a,b):
    return a*b

def sub (a,b):
    if a >= b:
     return a-b
    return b-a

def div(a,b):
    if a%b == 0:
        return a//b
    return False
```

4. Puisque `v = op(valeurs[i], valeurs[j])`, alors `v` contient la valeur de `sub(7,10)`, soit `3`.
5. - La clé primaire de la relation `joueur` est l'attribut `id`.
   - Les clés étrangères de la table `partie` sont les attributs `gagnant` et `perdant` qui font référence à l'attribut `id` de la table `joueur`.
6. La requête suivante convient :

```
SELECT nom FROM joueur
JOIN partie ON partie.gagnant = joueur.id
WHERE idP = 1
```

7. Il faut d'abord supprimer de la table `partie` les enregistrements qui font référence à cet enregistrement de la table `joueur` afin qu'il n'y ait pas violation de contrainte d'intégrité :

```
DELETE FROM partie WHERE gagnant = 9 OR perdant = 9
```

   On peut alors supprimer l'enregistrement en question de la table `joueur` :

```
DELETE FROM joueur WHERE id = 9
```

8. La requête suivante convient :
   - En utilisant l'identifiant :

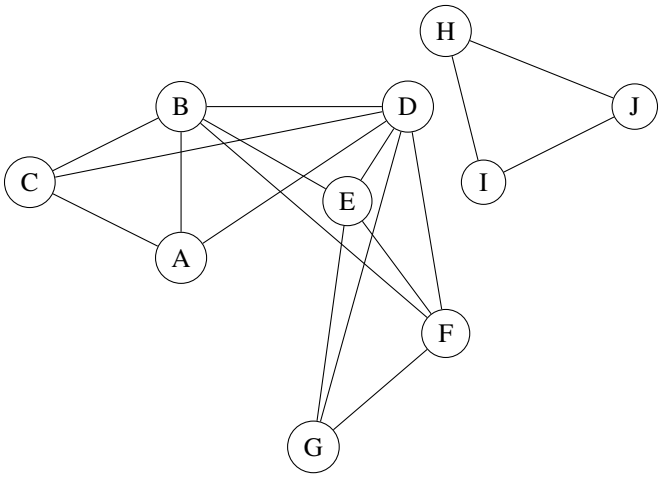
```
SELECT COUNT(*) FROM partie WHERE gagnant = 8
```

   - Sans utiliser l'identifiant :

```
SELECT COUNT(*) FROM partie
WHERE gagnant = (SELECT id FROM joueur WHERE nom='Renard' AND prenom='Bertand')
```

**Exercice 2 (Graphe et POO)**
**Partie A**

1. Par exempe, les personnes A et B sont amies.
2. Par exemple, les personnes A et D ne sont pas amies, mais ont B comme ami commun.
3. On a le graphe suivant :




4. Il suffit de deux jours pour que A et E deviennent amis.
5. Il n'y aura plus de recommandation au bout du troisième jour.

**Partie B**

6. La fonction `mystere` indique si `A` et `B` sont amis ou non.
7. La fonction suivante convient :

```
def amis_commun(A, B):
    for S in A.voisins:
        if B in S.voisins:
            return True
    return False
```

8. On a les codes suivants :
   - Ligne 9 : `L.append(S.nom)`
   - Ligne 11 : `if E.couleur == 'bleu':`
   - Ligne 12 : `parcours(E, L)`
   - Ligne 18 : `if B.nom in parcours(A):`
9. Dans un parcours en largeur, c'est une file qu'il faut utiliser. La fonction suivante convient :

```
def parcours_largeur(A):
    for N in G.noeuds:
        N.couleur = 'bleu'
    f = File()
    f.enfiler(A)
    A.couleur = 'rouge' # sommet vu
    while not f.est_videF():
        S = f.defiler()
        print(S.nom)
        for E in S.voisins:
            if E.couleur == 'bleu': # sommet jamais vu
                f.enfiler(E)
                E.couleur='rouge' # sommet désormais vu
```

**Exercice 3 (Réseau, protocoles de routage et algorithmique)**
**Partie A : adressage d'un réseau**

1. En utilisant les rappels de l'énoncé, on en déduit que le masque est `11111111.11111111.11111111.00000000` en binaire, soit `255.255.255.0` en décimal.
2. En utilisant les rappels de l'énoncé, on a directement que l'adresse du réseau est `172.21.200.0`
3. En utilisant les rappels de l'énoncé, on a directement que l'adresse de diffusion est `172.21.200.255`
4. En utilisant les rappels de l'énoncé, on déduit facilement que le nombre d'adresses disponibles est $256 - 2 = 254$ (il faut retirer les deux adresses correspondant respectivement au réseau et au broadcast).

**Partie B : tri par comptage**

5. On a la liste suivante : `[0, 1, 0, 1, 3, 0, 1, 0, 2, 3]`
6. La fonction suivante convient :

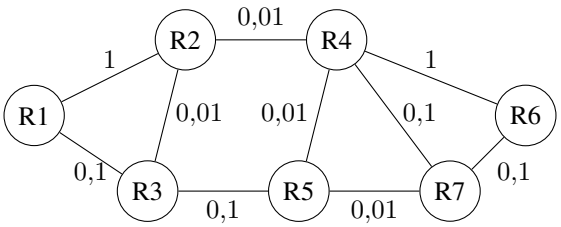
```
def maximum(tab):
    m = 0 # initialisation possible car tableau de nombres positifs
    for v in tab :
        if v > m :
            m = v
    return m
```

7. Le code suivant convient :

```
nouv_tab = [0]*len(tab)
position = 0
for i in range(len(liste_compteurs)):
    val = liste_compteurs[i]
    for j in range(val):
        nouv_tab[position] = i
        position = position + 1
return nouv_tab
```

**Partie C : protocoles de routage**

8. On peut choisir `10.1.1.2` pour l'interface 1 et `10.1.3.1` pour l'interface 2.
9. On traverse successivement les routeurs R1, R2, R4 et R6.
10. On a deux routes possibles :
    - PC2 - R1 - R2 - R4 - R7 - R6 - PC3
    - PC2 - R1 - R3 - R5 - R7 - R6 - PC3
11. Puisque 10 Gbit/s = $10^{10}$ bit/s, on en déduit que le coût est de $1/100$, soit 0,01.
12. On a le schéma suivant :




13. On obtient le chemin R1 - R3 - R2 - R4 - R5 - R7 - R6 pour un coût total de 0,24.

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.