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

22NSIJ2AN1

Exercice 1

1. (a) La clé primaire de la relation Sport est le couple (nomSport,nomStation), et une clé étrangère est nomStation.

   (b) Une contrainte d'intégrité de domaine est que prix doit être un entier, une contrainte d'intégrité de relation est que la clé primaire (nomSport,nomStation) doit être unique, une contrainte d'intégrité de référence est que nomStation doit faire référence à une station présente dans la table Station.

2. (a) La requête INSERT INTO est rejetée car la clé primaire ("planche à voile","La tramontane catalane") est déjà présente dans la table, il s'agit d'une violation de contrainte de relation.
   Il faut ici utiliser une requête UPDATE, plus précisément UPDATE Sport SET prix=1350 WHERE nomSport='planche à voile' ;

   (b) Pour respecter les contraintes d'intégrité de référence, il faut commencer par insérer la station dans la table Station, puis le sport dans la table Sport.
   INSERT INTO Station VALUES ("Soleil Rouge","Bastia","Corse") ;
   INSERT INTO Sport VALUES ("plongée","Soleil Rouge",900) ;

3. (a) SELECT mail FROM Client ; permet d'obtenir les mails de tous les clients ; on peut aussi utiliser SELECT DISTINCT mail FROM Client ; pour éviter les doublons et ne pas envoyer plusieurs fois le même message à une boîte mail servant à plusieurs clients.

   (b) SELECT nomStation FROM Sport WHERE nomSPort="plongée" ; permet d'obtenir les noms des stations où on peut pratiquer la plongée.

4. (a) SELECT ville, Station.nomStation FROM Station JOIN Sport
   ON Station.nomStation = Sport.nomStation WHERE nomSport="plongée" ;
   permet d'obtenir les noms des villes et des stations où on peut pratiquer la plongée.

   (b) Le nombre total de séjours effectués en Corse en 2020 peut être obtenu avec la fonction d'agrégation COUNT et la requête SELECT COUNT(*) FROM Sejour JOIN Station
   ON Sejour.nomStation=Station.nomStation WHERE Station.region="Corse"
   AND Sejour.annee=2020 ;

Exercice 2

1. En utilisant les tables de routage, le chemin emprunté par un paquet de R2 à R7 est R2-R1-R4-R7, et l'accusé de réception empruntera le chemin R7-R4-R3-R2.

2. (a) En cas de panne du routeur R4, les routeurs R1, R2 et R3 ne sont lus connectés aux routeurs R5, R6 et R7.

   (b) Pour remédier à ce problème on peut ajouter un lien, par exemple entre R1 et R6.

3. On a refait le graphe représentant le réseau, en ajoutant R8 et les liaisons R8-R2 et R8-R6. On en déduit la table de routage de R8 et une nouvelle table de routage pour R2.

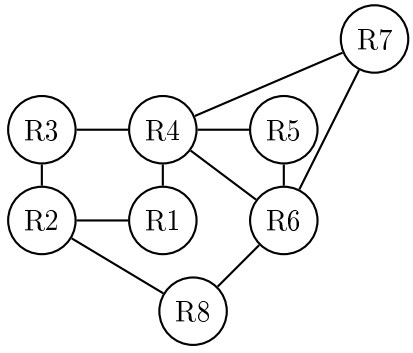


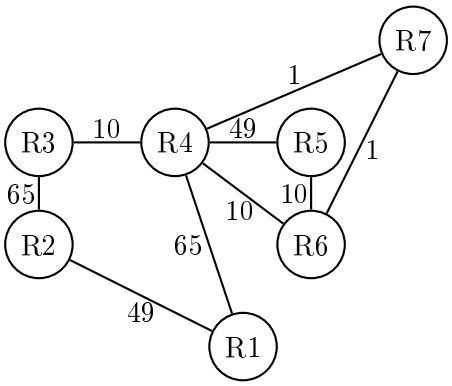
table de routage de R8		
Destination	Lien	Distance
R1	R2	2
R2	R2	1
R3	R2	2
R4	R6	2
R5	R6	2
R6	R6	1
R7	R6	2

table de routage de R2		
Destination	Lien	Distance
R1	R1	1
R3	R3	1
R4	R1	2
R5	R3	3
R6	R8	2
R7	R1	3
R8	R8	1

4. (a) La bande passante de Fast Ethernet se calcule à partir de son coût, $BP = \dfrac{10^8}{1} = 100$ Mbit/s.

   Le coût du réseau de type Ethernet est $C = \dfrac{10^8}{10 \times 10^6} = 10$.

(b) Pour déterminer le chemin le moins coûteux de R2 à R5, on a reproduit ci-dessous le graphe **pondéré** par les coûts.



Le chemin le moins coûteux pour relier R2 à R5 est R2-R3-R4-R7-R6-R5, dont le coût est 87.

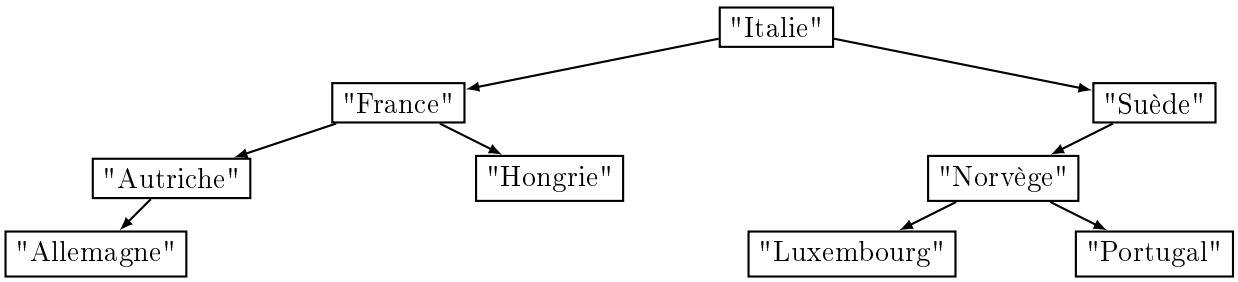
On peut trouver ce chemin grâce à l'algorithme de Dijkstra, ou bien remarquer que tout chemin de R2 à R7 doit d'abord joindre R2 à R4 puis R4 à R7 ; pour aller de R2 à R4 il y a deux chemins, et le plus court est R2-R3-R4 ; pour aller de R4 à R5, après R4 on a trois possibilités, R5, R6 ou R7, et passer par R6 ou aller directement à R5 est plus coûteux que de passer par R7 puis R6.

Exercice 3

1. (a) La hauteur de cet arbre est 3, en utilisant la définition imposée par l'énoncé.

   (b) La valeur booléenne de l'expression `"Allemagne"<"Portugal"` est `True`.

   (c) L'ajout de nouveaux nœuds dans l'ABR se fait au niveau des feuilles.



2. Le parcours en largeur de l'arbre de l'énoncé est Italie, France, Suède, Autriche, Hongrie, Norvège, où les niveaux sont parcourus du haut vers le bas et de gauche à droite.

3. Avec les primitives proposées par l'énoncé pour opérer sur des arbres binaires, on peut compléter la fonction ainsi :

```
def recherche(arb,val):
    """ cette fonction booléenne cherche la présence de val dans l'ABR arb """
    if est_vide(arb):
        return False
    if val == racine(arb):
        return True
    if val < racine(arb):
        return recherche(gauche(arb),val)
    else:
        return recherche(droite(arb),val)
```

On reconnaît l'algorithme de la dichotomie.

4. La fonction ci-dessous calcule récursivement la taille d'un arbre.

```
def taille(arb):
    if est_vide(arb):
        return 0
    return 1+taille(gauche(arb))+taille(droite(arb))
```

Exercice 4

1. (a) La bonne proposition est la proposition 3, puisqu'une chaîne de longueur 0 ou 1 est nécessairement un palindrome.

```
1  if len(txt)<2:
2      return True
```

   (b) Lors de l'appel `palindrome("bonjour")`, ligne 9 on a :

txt[0]	txt[taille-1]	interieur
"b"	"r"	"onjou"

2. Pour une fonction booléenne, il faut fournir au moins un cas d'usage devant renvoyer `True` et un cas d'usage renvoyant `False`, p.ex `palindrome("laval")` et `palindrome("bonjour")`.

3. Une version dérécursifiée du même algorithme :

```
1  def palindrome(txt):
2      n=len(txt)
3      for k in range(n//2):
4          if txt[k]!=txt[n-1-k]:
5              return False
6      return False
```

4. (a) On écrit la fonction demandée.

```
1  def complementaire(txt):
2      s=''
3      for lettre in txt:
4          if lettre=='A':
5              s=s+'T'
6          if lettre=='T':
7              s=s+'A'
8          if lettre=='C':
9              s=s+'G'
10         if lettre=='G':
11             s=s+'C'
12     return s
```

   (b) `'GATCGT'` est palindromique si `'GATCGT'+'CTAGCA'` est un palindrome, ce qui n'est pas le cas car `'G'!='A'`.

   (c) La fonction ci-dessous utilise les fonctions déjà écrites pour déterminer si une chaîne est palindromique.

```
1  def est_palindromique(txt):
2      return palindrome(txt+complementaire(txt))
```

Exercice 5

1. (a) La bonne proposition est la proposition 2.

$\longrightarrow$		
14	17	15

   (b) Cette file peut être créée avec la séquence d'instructions `F=creer_file_vide()`, `enfiler(F,15)`, `enfiler(F,17)`, `enfiler(F,14)`.

2. L'algorithme est de vider la file dans une autre file en comptant les éléments, puis de vider la deuxième file dans la première pour la restaurer dans son état initial.

```python
def longueur_file(F):
    """File -> Int"""
    G=creer_file_vide()
    n=0
    while not est_vide(F):
        enfiler(F,defiler(G))
        n=n+1
    while not est_vide(G):
        enfiler(G,defiler(F))
    return n
```

3. On complète la fonction `variations`.

```python
def variations(F):
    """ File -> Tableau """
    taille=longueur_file(F)
    if taille==1:
        return []
    else:
        tab=[0 for k in range(taille-1)]
        element1=defiler(F)
        for i in range(taille-1):
            element2=defiler(F)
            tab[i]=element2-element1
            element1=element2
        return tab
```

   Le fait qu'à chaque tour de boucle `element1` et `element2` désignent des éléments successifs de la file est une technique très classique qui s'appelle **fenêtre glissante** (sliding window).

4. On fournit la fonction demandée.

```python
def nombre_baisses(tab):
    n=0
    m=0
    for x in tab:
        if x<0:
            n=n+1
        if x<m:
            m=x
    return n,m
```

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.