

## Exercice 1

- 1) Il est le symbole pour obtenir le quotient de la division entière. Donc  $7//2$  vaut 3.  
 Donc  $x$  et  $y$  ont la même valeur 3. Donc  $x == y$  est Vrai.  
la réponse est True
- 2) Le bit de poids faible des premiers, troisièmes et quatrièmes nombres est 1. Donc ils sont impairs. Donc le seul nombre pair est le deuxième.  
la réponse est 100010
- 3) Fournis la table de vérité de  $(P \text{ et } Q)$  ou  $(\text{non}(P) \text{ ou } Q)$
- | P | Q | $P \text{ et } Q$ | $(\text{non}(P) / \text{non}(P) \text{ ou } Q)$ | $(P \text{ et } Q)$ ou $(\text{non}(P) \text{ ou } Q)$ |
|---|---|-------------------|---|--|
| F | F | F                 | V   | V  |
| F | V | F                 | V   | V  |
| V | F | F                 | F   | F  |
| V | V | V                 | F   | V  |
- la réponse est False
- 4) Parmi les quatre listes, 14 est présent dans la liste  $[11, 12, 13, 14, 15]$   
 c'est à dire  $L[2]$   
 14 est l'élément d'indice 3 dans cette liste.  
 Donc l'expression  $L[2][3]$  vaut 14.  
la réponse est  $L[2][3]$
- 5) liste =  $[1, [2, 3], [4, 5], 6, 7]$ . Liste contient 5 éléments. Sa longueur est 5.  
 $\begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ 1^{\text{er}} \text{élément} & 2^{\text{e}} & 3^{\text{e}} & 4^{\text{e}} & 5^{\text{e}} \end{matrix}$   
la réponse est 5
- 6) La fonction foo contient une boucle for qui parcourt toute la liste en comparant chaque élément  $\text{liste}[i]$  avec son suivant. Dès qu'on rencontre le cas  $\text{liste}[i] > \text{liste}[i+1]$ , c'est à dire dès qu'on trouve un élément qui est strictement supérieur à son suivant, alors la fonction renvoie False.  
 Dans le cas contraire (aucun élément supérieur à son suivant) alors la fonction renvoie True. Autrement dit la liste est triée croissante.  
 Parmi les listes proposées, seule la dernière est triée croissante.  
la réponse est  $\text{foo}([1, 2, 2, 3, 4, 5])$
- 7) lst est une liste en compréhension qui contient les éléments présents dans variable =  $[8, 12, -7, 52, -5, 32]$  si l'élément est strictement inférieur à 12.  
 Donc lst vaut  $[8, -7, -5]$ .  
la réponse est  $[8, -7, -5]$
- 8) fruit est initialisé comme étant un dictionnaire vide.  
 La fonction addone est appelée 5 fois avec comme paramètre le dictionnaire fruit.  
 A chaque appel, on lui fournit en premier paramètre une clé qui est le nom d'un fruit.  
 Si cette clé est dans le dictionnaire alors la valeur de l'item augmente de 1 (c'est l'instruction  $\text{dic}[\text{index}] += 1$ ).  
 Sinon l'item est créé avec la valeur 1 (c'est l'instruction  $\text{dic}[\text{index}] = 1$ ).  
 Donc la réponse est  $\{\text{'Apple': 1, 'Banane': 1, 'Orange': 3}\}$

## Exercice 2

- 1) Le schéma 1 montre deux inconvénients de l'envoi en flux continu:
  - Si le fil est coupé alors le flux est interrompu.
  - Si un routeur est occupé alors le flux est bloqué.
- 2) Le schéma 2 montre deux avantages du découpage en paquets:
  - Si un paquet est perdu, la totalité du message ne doit pas être renvoyée à nouveau. Seulement le paquet perdu doit être renvoyé à nouveau.
  - Le routeur n'est pas occupé exclusivement pour un flux. Donc les flux ne sont pas bloqués.
- 3) Le travail de la couche de transport du modèle OSI est
  - de découper les données en paquets
  - d'ajouter à chaque paquet une en-tête de transport.

### Exercice 3

#### Partie 1

1) les descripteurs de la table sont:  
code\_département, nom\_département, code\_région, nom\_région

2) L'induction est:  
table\_départements = lecture\_fichier('départements.csv')

3) la variable table\_départements est du type liste de listes.

4) la fonction recherche\_1 est appelée avec comme deuxième paramètre la fonction qui donne la valeur booléenne de  $x[2] == '52'$

Dans le if de la fonction, on a comme condition critère (table[i]) c'est à dire table[i][2] == '52'

la condition est vraie lorsque la ligne table[i] contient '52' comme élément d'index [2] c'est à dire lorsque le département examiné a un code de région '52'.

Donc la variable recherche\_1 va contenir tous les départements des pays de la Loue.

5) recherche\_2 = sélectionner(table\_départements, lambda x: int(x[2]) < 10)

6) recherche\_3 = sélectionner(table\_départements, lambda x: x[2] == '28' or x[2] == '53')

#### Partie 2

1) les\_descripteurs = table\_communes[0]

2) s1 contient les communes classées par ordre du nom commune croissant.

3) s2 contient les communes classées par ordre de latitude décroissante.

4) ma\_sélection = s2[0:5]

5) a) s3 = sorted(table\_communes[1:], key=lambda ligne: ligne[1])

b) [ligne for ligne in s3 if ligne[1] == '53']

## Exercice 4

1) niveau2[2] = 1  
niveau2[3] = 2

```
2) def conforme(niveau):  
    for i in range(len(niveau)-1):  
        if niveau[i+1] - niveau[i] > 1:  
            return False  
        elif niveau[i+1] - niveau[i] < -2:  
            return False  
    return True
```

```
3) def difficulte(niveau):  
    d = 0  
    for element in niveau:  
        d = d + element  
    return d
```

4) niveau3[8][1] = 1

```
5) def score(niveau, x):  
    score = 0  
    for i in range(0, x+1):  
        score = score + niveau[i][1]  
    return score
```

6) meilleurs\_scores['Alice'] = 32

7) la variable  $i$  de la boucle `for i in dico` prend à chaque tour de boucle la clé des items présents dans dico.

Si la valeur `dico[i]` de l'item ayant pour clé  $i$  est strictement supérieure à la valeur précédente de  $x$  alors  $x$  est mis à jour avec la valeur `dico[i]` et  $y$  est mis à jour avec la clé.

En fin de boucle, la valeur de la clé  $y$  est renvoyée. C'est donc la clé qui correspond à l'item de plus grande valeur.

Conclusion: la fonction renvoie le nom de celui qui a le score le plus élevé.