

## 2021 Polynésie sujet 2

### Exercice 1

#### Partie A

1. [0.25 point]

8  
[8, 7, 18, 16, 12, 9, 17, 3]

2. [0.25 point]

```
for i in range(2, 5):
    print(notes[i])
```

#### Partie B

1. [0.25 point]

```
while i >= 0 and liste[i] > element_a_inserer:
    liste[i+1] = liste[i]
```

2. [0.25 point] [7, 8, 18, 14, 12, 9, 17, 3]

3. [0.5 point] 2<sup>e</sup>: [7, 8, 18, 14, 12, 9, 17, 3]  
3<sup>e</sup>: [7, 8, 14, 18, 12, 9, 17, 3]

#### Partie C

1. [0.25 point] Il est récursif car il s'appelle lui-même.

2. [0.5 point] On prend la plus petite des deux cartes visibles et on l'ajoute à la suite de celles qui sont dans notre main. Quand un tas est vide, on met l'autre tas à la suite des autres cartes de notre main.

3. [0.25 point]

```
tri_fusion(liste, i_debut, i_partage)
tri_fusion(liste, i_partage, i_fin)
fusionner(liste, i_debut, i_partage, i_fin)
```

4. [0.25 point] On importe le module math pour avoir accès à la fonction floor.

#### Partie D

1. [0.25 point] Tri fusion car on voit les étapes de fusion des listes.

2. [0.5 point]

Tri par insertion :  $O(n^2)$

Tri fusion :  $O(n \log_2(n))$

**3. [0.5 point]**

Tri par insertion : on parcourt deux fois la liste. Une première fois pour choisir l'élément à insérer et une deuxième fois pour choisir où l'insérer.

Tri fusion : la taille de la liste est divisée par deux à chaque étape. Le  $\log_2(n)$  vient de là.

## Exercice 2

### Partie A

**1. [0.25 point]**

Clients : IdClient

Articles : IdArticle

**2. [0.25 point]**

email : VARCHAR(50) (chaîne de caractères)

quantite : INT (entier)

**3. [0.25 point]** FOREIGN KEY (idClient) REFERENCE Clients(IdClient);

### Partie B

**1. [0.25 point]** Avec GET les données apparaissent dans l'URL qui est limitée en taille. Il est donc possible que les données ne rentrent pas.

**2. [0.25 point]** HTTPS car la communication sera alors chiffrée et ne pourra être lue par un tiers.

**3. [0.25 point]** Cela limite le nombre de requêtes et empêche des tentatives de piratage.

### Partie C

**1. [0.5 point]**

```
SELECT IdArticle, Libelle  
FROM Articles  
WHERE PrixEnCentimes <= 1500;
```

**2. [0.25 point]** Elle liste les commandes avec le client, son email et l'adresse de livraison dont le paiement n'est pas valide.

**3. [0.5 point]**

```
SELECT Libelle  
FROM Articles  
JOIN ArticlesCommande  
ON Articles.IdArticle = ArticleCommande.IdArticle  
WHERE IdCmd = 1345;
```

**4. [0.5 point]**

```
INSERT INTO Articles (Libelle, Description, PrixEnCentimes)  
VALUES ("Imperméable", "Cet imperméable se replie en forme de pochette", 999);
```

## Partie D

1. [0.5 point]

Dans Articles : stock : INT

Dans Clients : AdresseDefaut : VARCHAR(90)

2. [0.25 point]

Il faut remplacer la ligne 5 par :

```
Stock <- Stock - Quantite
```

## Exercice 3

### Partie A

1. [0.5 point]

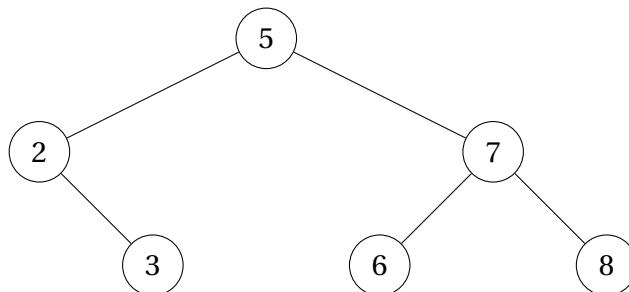
Racine : 5

Fils gauche : 2

Fils droit : 7

2. [0.5 point] 5, 2 et 3

3. [0.5 point]



### Partie B

1. [0.5 point] C'est un constructeur qui permet d'initialiser les attributs de l'instance.

2. [0.5 point] Il ne se passe rien.

3. [0.5 point]

```
arbre = ABR(5)
arbre.insereElement(2)
arbre.insereElement(3)
arbre.insereElement(7)
arbre.insereElement(8)
```

### Partie C

1. [0.5 point] Parcours infixe.

2. [0.5 point] Elle est en  $O(n \log(n))$  alors que les autres sont en  $O(n^2)$ . Elle est plus efficace.

## Exercice 4

### Partie A

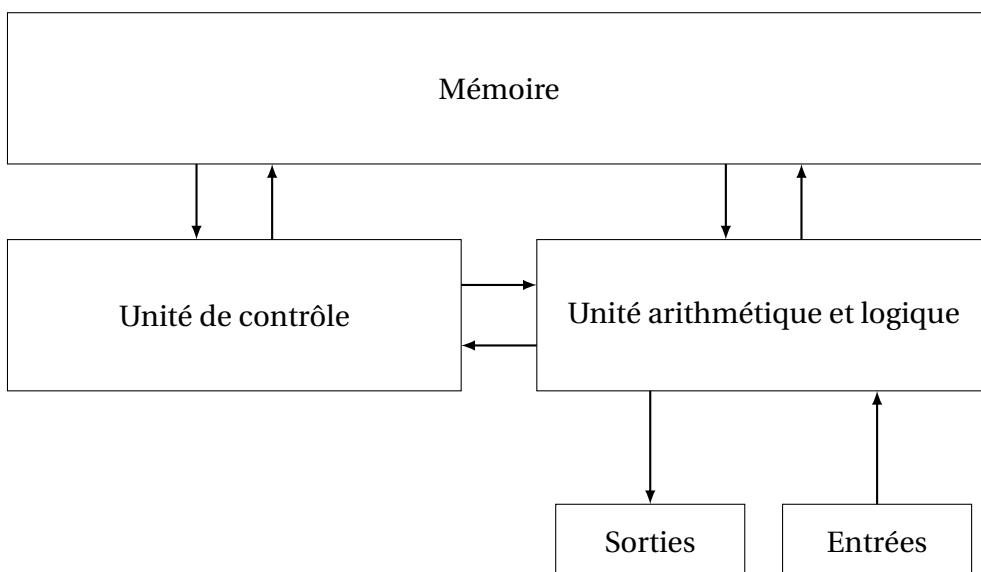
1. [0.25 point] Pour permettre une détection et une correction plus facile des erreurs.
- 2.a. [0.25 point] Les sommets sont les routeurs ou ordinateurs. Les arrêtes sont les liaisons entre ces éléments.
- 2.b. [0.25 point] Le nombre de sauts.

### Partie B

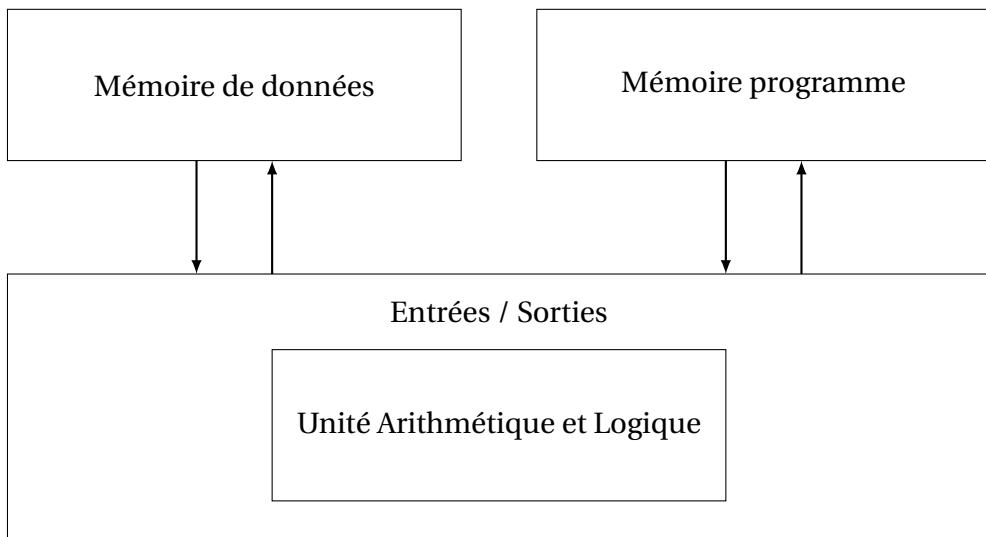
1. [0.5 point] On suppose que l'on a deux processus A et B bloquant respectivement les ressources c et d. Si B attend c et A attend d, il y a interblocage.
2. [0.25 point] On peut veiller à ce qu'un processus libère une ressource avant d'en attendre une autre.

### Partie C

1. [0.25 point]



2. [0.5 point]  
Le CP est situé dans l'unité de contrôle, il contient l'adresse mémoire de l'instruction en cours d'exécution.  
Le RI est aussi situé dans l'unité de contrôle, il contient l'instruction en cours d'exécution.
3. [0.25 point]



- 4.** [0.5 point] Une mémoire morte se conserve sans alimentation contrairement à une mémoire vive. La mémoire programme est une mémoire morte dans un microcontrôleur car il faut la conserver lorsqu'on coupe l'alimentation.

## Partie D

1. [0.25 point] On peut effectuer plusieurs tâches en parallèle.
2. [0.25 point] Les vitesses d'échange avec les entrées / sorties peuvent être différentes de ceux avec la mémoire.
3. [0.25 point] Les échanges entre les éléments sont plus rapides.
4. [0.25 point] Il est impossible de changer un seul élément en cas de panne, il faut tout changer. La puissance de calcul est généralement plus faible.

## Exercice 5

### Partie A

1. [0.25 point]  
Films : IdFilm  
Abonnes : IdAbonne

2. [0.25 point]  
IdFilm : INT (entier)  
Description : VARCHAR(150) (chaîne de caractère)

3. [0.25 point]  
Primaire : IdCpt  
Etrangère : IdAbonne

4. [0.25 point] On crée une table Acteurs avec IdActeur comme clé primaire, Nom pour le nom de l'acteur et IdFilm une clé étrangère en lien avec le film.

5. [0.5 point] On ajoute l'attribut AgeMin (INT) à la table Films et AgeMin et AgeMax à la table ComptesAbonnes.

### Partie B

1. [0.25 point]

```
SELECT IdCpt, Pseudo  
FROM ComptesAbonnes  
WHERE IdAbonne = 237;
```

2. [0.25 point] Elle donne le nombre moyen d'étoiles du film avec l'IdFilm 1542.

3. [0.5 point] Elle renvoie l'IdFilm, le titre et le nombre d'étoiles de tous les films du compte 508 classés par nombre d'étoiles décroissant.

4. [0.25 point]

```
UPDATE ComptesAbonnes  
SET pseudo = "Champion"  
WHERE IdCpt = 508;
```

## Partie C

1. [0.5 point] Elle donne l'écart moyen entre les notes des deux utilisateurs IdCpt1 et IdCpt2 pour les films de la liste listeFilms

2. [0.75 point]

```
def conseilsFilms(IdCpt):  
    conseils = []  
    bestFilms = podium(IdCpt)  
    listSpectateurs = spectateurs(bestFilms)  
    for spectateurs in listSpectateurs:  
        if distance(spectateur, idCpt, bestFilms) < 10:  
            bestFilmsSpect = podiumCompte(spectateur)  
            for i in range(min(3, len(bestFilmsSpect))):  
                conseils.append(bestFilmsSpect[i])  
    return conseils
```