

<b>EXAMEN SEMESTRIEL FONDEMENTS DES BASE DE DONNEES</b> <b>SECTION L1SI 1, 2,3&amp;4</b> <b>Durée 1h30</b> <b>ANNEE UNIVERSITAIRE : 2022-2023</b> <p>Documents non autorisés : Tous documents du cours/td/tp, notes manuscrites (nb : pas de livres)</p>	
	<b>PROF M<sup>e</sup> J.RAOUF</b>

## Exercice N°1 (QCM)

Cocher la (les) réponse(s) correcte(s)

**1. SGBD signifie :**

- Service de gestion des bases de données
- Système de gestion des bases de données
- Système de gestion des bandes de données

**2. Un SGBD est un :**

- Logiciel de gestion de stock
- Fichier de gestion des bases de données
- Logiciel de gestion des bases de données

**3. Une base de données est :**

Ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur.

Ensemble de données avec un objectif commun.

Fichier de données avec un objectif commun.

Une base de données peut être :

Relationnel

En réseau

En anneau

**7. Un administrateur de base de données a pour rôle :**

La définition, la description et la création de la base.

La gestion de la base.

L'installation d'un SGBD.

**8. Les fonctions d'un SGBD sont :**

LLD, LMD, LDM

LMD, LCD, LDD

LDD, LMD, LDC

**9. LMD signifie :**

- Langage de modélisation des données
- Langage de manipulation des données
- Liste de modèle de données

**10. Le langage de Contrôle de Données permet :**

- La connexion et l'utilisation de la base de données.
- La création de la base de données.
- Le contrôle et la sécurité des données.

**11. Les niveaux de description des données dans un SGBD sont :**

Niveau externe, niveau création, niveau interne.

Niveau haut, niveau bas, niveau au milieu.

Niveau externe, niveau conceptuel, niveau interne.

**12. Le niveau qui prend en charge le stockage des données est :**

- Niveau externe,
- Niveau conceptuel,
- Niveau interne.

**13. Le niveau qui prend en charge le stockage des données est :**

- Niveau externe,
- Niveau conceptuel,
- Niveau interne.

**13. Le niveau qui prend en charge le problème du dialogue avec les utilisateurs est :**

- Niveau externe,
- Niveau conceptuel,
- Niveau interne.

**14. Le niveau qui prend en charge la description de la structure de toutes les données est :**

- Niveau externe,
- Niveau conceptuel,
- Niveau interne.

**15. Le niveau qui prend en charge la description de la structure de toutes les données est :**

- Niveau externe,
- Niveau conceptuel,
- Niveau interne.

## **15. Le niveau conceptuel de l'architecture ANSI/SPARC**

permet :

Le stockage des données.

La création de la base de données.

La description de la structure des données.

## **16. Le modèle qui nécessite la spécification d'un SGBD est :**

Modèle physique

Modèle conceptuel

Modèle logique

## **17. La modélisation qui représente les structures de stockage internes et détaille l'organisation des fichiers est :**

Modélisation physique

Modélisation logique

## **18. La modélisation qui traduit le modèle conceptuel dans le modèle du SGBD est :**

Modélisation physique

Modélisation conceptuel

Modélisation logique.

## **19. Dans le processus de conception d'une base de données, la conception physique c'est le :**

Passage du modèle logique au modèle conceptuel

Passage du modèle logique au modèle physique

Passage du modèle conceptuel au modèle logique

## **20. Dans le processus de conception d'une base de données, l'ordre de conception d'une base est :**

Modèle physique, Modèle conceptuel, Modèle logique

Modèle physique, Modèle conceptuel, Modèle logique

Modèle conceptuel, Modèle logique, Modèle physique

## **Exercice N°2**

### **I. LDD & LMD (Langage définition et manipulation des données)**

Soit un schéma relationnel composé de la relation

Frs (numf, ville) et la relation

Article (code, prix, qte, #numf)

On propose l'extension suivante des relations suivantes :

Article			
Code	prix	qte	numf
A1	1200	20	F2
A2	3200	100	F3
A3	450	50	F2

Frs	
NumF	Ville
F1	SBZ
F2	Sfax
F3	SBZ

1. Donner la requête SQL correspondante à la création de la table Article. (code, numf, ville : des chaînes de caractères de taille maximale 30, prix et qte des entiers de taille maximale 20)

2. Donner la requête SQL correspondante à l'insertion des enregistrements de la table Frs

3. Donner la commande SQL pour augmenter la quantité des Article de 10 du fournisseur F2.

4. Donner la commande SQL pour afficher le nombre des articles fournis par le fournisseur F2.

5. Donner la commande SQL pour supprimer les articles de numéro A2.

6. Donner la commande SQL pour supprimer la table Article et la table Frs (Respectez l'ordre).

### **II. LID (Langage manipulation des données)**

Soit le schéma de base de données relationnel suivant :

AGENCE (Num\_Agence, Nom, Ville)

CLIENT (Num\_Client, Nom, Ville)

COMPTE (Num\_Compte, #Num\_Agence, #Num\_Client, Solde)

EMPRUNT (Num\_Emprunt, #Num\_Agence, #Num\_Client, Montant)

1. Ecrire les requêtes suivantes en SQL:

2. Les noms et les villes des Agences.

3. Les montants des emprunts  $\geq 25000$ .

4. Liste des clients dont la ville ne contient pas le caractère 's'.

5. Les numéros des emprunts ayant un montant supérieur à 12000.

6. Les noms des clients qui habitent "Sidi Bouzid".

7. Les montants des emprunts des clients numéro 12 et 13.

8. Le nombre des clients qui n'ont pas des emprunts.

9. Le montant maximum, minimum et la moyenne des soldes des Comptes.

10. La liste des agences ayant des comptes-clients.

11. Les Clients ayant un compte à une agence à Paris.

Ministère  
de l'Enseignement Supérieur  
et de la Recherche Scientifique

Université de Gabès  
Institut Supérieur  
de l'Informatique de Médenine

Filière : L1SI  
Année Universitaire : 2022-2023  
Nbre de pages : 1

Enseignant : JARRAY A.  
Date : 10-05-2022  
Durée De L'épreuve : 1H30mn

### ALGÈBRE I

*NB : Il sera tenu compte de la présentation des copies et de la bonne rédaction.*

Calculatrice et Documents non autorisés

#### Exercice : 01

Soit  $\mathcal{B}_c = (e_1, e_2, e_3)$  la base canonique de  $\mathbb{R}^3$ .  
 $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  une application linéaire définie dans la base  $\mathcal{B}_c$  par la matrice :

$$A = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

1. Calculer  $\det(A)$ , En déduire le rang de  $f$ .
2. Montrer que  $\text{Ker}(f) \neq \{0\}$ . et en déduire que 0 est une valeur propre de  $A$
3. Calculer le polynôme caractéristique  $P_A$  de  $A$ . Déduire que les autres valeurs propres de  $A$  sont 1 et 2.
4. En déduire que  $A$  est diagonalisable.  
On définit les trois vecteurs propres  $u$ ,  $v$  et  $w$  par :  
 $E_0 = \text{vect}(u)$  ;  $E_1 = \text{vect}(v)$  et  $E_2 = \text{vect}(w)$   
avec  $E_0$ ,  $E_1$  et  $E_2$  sont les sous espaces propres associés aux valeurs propres 0, 1 et 2.
5. Déterminer les vecteurs  $u$ ,  $v$  et  $w$ .
6. Montrer que  $B = (u, v, w)$  est une base de  $\mathbb{R}^3$ .
7. En déduire la matrice  $D$  de  $f$  dans la base  $B$ .
8. Calculer  $P$  la matrice de passage de la base  $\mathcal{B}_c$  à la base  $B$ .
9. En déduire  $P^{-1}$ .
10. Calculer  $D^n$ .
11. Donner une relation entre  $A$  et  $D$  et Calculer  $A^n$

#### Exercice : 02

Soit  $A = \begin{pmatrix} 1 & 1 & -m \\ 1 & 1 & 0 \\ 0 & 0 & m \end{pmatrix}$  où  $m \in \mathbb{R}$

1. Pour quelles valeurs de  $m$   $A$  est inversible.
2. Calculer le polynôme caractéristique  $P_A$  de  $A$ , déduire que  $A$  est trigonalisable.
3. Déterminer suivant les valeurs du paramètre  $m$  les valeurs propres de  $A$  et l'ordre de multiplicité de chaque valeur propre.
4. Déterminer les valeurs de  $m$  pour lesquelles  $A$  est diagonalisable.

*Bon Travail*

## *Examen Algorithmique et Structure de données*

Durée : 1 h 30 mn

Documents non autorisés

### **Exercice 1**

- 1) Une pile P1 contient une suite d'entiers. Écrire la procédure **PilePos (var P1 : Pile)** qui supprime les entiers négatifs de P1 et laisse dans P1 les entiers positifs. Vous pouvez résoudre le problème à l'aide d'une File temporaire.
- 2) Écrire une procédure **DéfilerNelt(var F : File ; N : entier)** qui permet de défiler N éléments si la file contient au moins N éléments, sinon elle défile toute la file.

On suppose que l'on dispose des primitives suivantes:

- **InitialiserPile(var P : Pile)** qui permet de créer une pile vide.
- **EstPileVide(P : Pile) :Booléen** qui permet de vérifier si une pile est vide.
- **Empiler(x : Entier ; var P : Pile)** qui permet d'ajouter l'élément x au sommet de la pile.
- **Dépiler(var x : Entier ; var P : Pile)** qui permet de supprimer le sommet de la pile et de le mettre la valeur dans la variable x.
- **InitialiserFile(var F : File)** qui permet de créer une file vide.
- **EstFileVide(F : File)** qui permet de vérifier si une file est vide.
- **Enfiler(x : Entier ; var F : File)** qui permet d'ajouter l'élément x à la queue de la file.
- **Défiler(var x : Entier ; var F : File)** qui permet de supprimer le sommet de la file et de le mettre la valeur dans la variable x.

### **Exercice 2**

1) Dessiner l'arbre binaire de recherche obtenu par insertions successives des éléments de clés **32, 24, 39, 52, 18, 26, 35** en partant de l'arbre vide.

2) Écrire une fonction qui permet d'afficher les valeurs des nœuds d'un arbre binaire de recherche par ordre croissant.

- 3) Donner le résultat obtenu en utilisant le parcours suivant (**Parcours1**) de l'arbre:

```

Procédure Parcours1(B : Arbre)
Début
    Si B ≠ nil Alors
        Parcours1(B^.FilsG)
        Parcours1(B^.FilsD)
        Ecrire(B^.val)
    Fin Si
Fin

```

- 4) Écrire une fonction qui permet de chercher un élément x dans l'arbre et de renvoyer vrai si ce dernier est existant et faux si non
- 5) Écrire une fonction *EstUneFeuille(B :Arbre) : Booléen* qui renvoie vrai si le noeud de l'arbre n'admet aucun fils et faux si non.

### Exercice 3

- 1) Pour chacun des fonctions  $T_i(n)$  suivant, déterminer sa complexité asymptotique dans la notation Grand-O. Exemple :  $T_0(n) = 3n \in O(n)$ .
- $T_1(n) = 6n^3 + 10n^2 + 5n + 2$
  - $T_2(n) = 3 \log_2 n + 4$
  - $T_3(n) = 2^n + 6n^2 + 7n$
  - $T_4(n) = 7k + 2$
  - $T_4(n) = 4 n^2 + n$
  - $T_5(n) = 5n+9$
- 2) Considérer les deux algorithmes A1 et A2 avec leurs temps d'exécution  $T_1(n) = 3n^2$  et  $T_2(n) = 100n + 96$  respectives.
- Déterminer la complexité asymptotique des deux algorithmes dans la notation Grand-O.
  - Quel algorithme a la meilleure complexité asymptotique?



Matière : Systèmes d'Exploitation 2  
Enseignant : M. Faouzi HAJJEM  
Filière : L1 GLSI

A.U. : 2022/2023  
Durée : 1 H 30

## Examen Systèmes d'Exploitation II

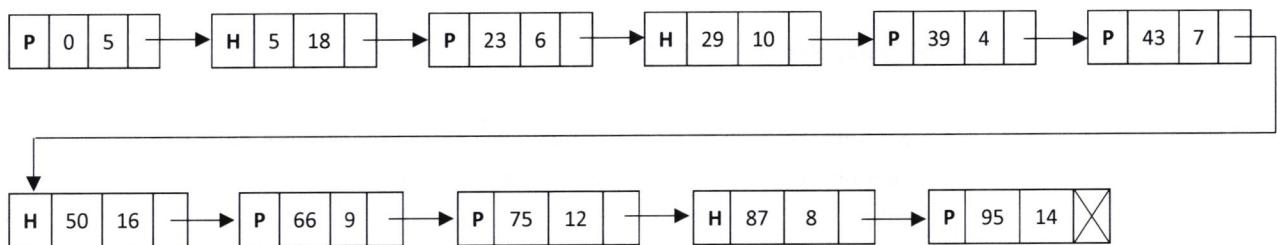
(Session Principale Mai 2023)

- L'usage de tout document et l'échange de matériels (document, stylo, règle, calculatrice, etc.) entre candidats sont strictement interdits.
- Les candidats ne sont pas autorisés à quitter la salle, temporairement ou définitivement sans remettre leurs copies, même blanches, et signer dans la liste d'émarginement.
- La présentation et la clarté de la feuille d'examen seront prises en compte.
- Cet examen comporte deux pages.

### Exercice 1 : Mémoire centrale [10 Pts]

Soit un système disposant d'une mémoire centrale utilisant une stratégie d'allocation basée sur des partitions variables. L'occupation de la mémoire RAM est gérée par une liste chaînée.

On suppose que l'état de la partition « User » de la RAM est décrit par le schéma suivant :



Soient les requêtes d'allocation et de libération de mémoire qui arrivent dans cet ordre :

- Arrivée d'un processus (10 Ko),
- Arrivée d'un processus (8 Ko),
- Fin du processus (4 Ko),
- Arrivée d'un processus (13 Ko),
- Fin du processus (9 Ko),
- Arrivée d'un processus (18 Ko),
- Fin du processus (14 Ko),
- Arrivée d'un processus (20 Ko).

**Travail à faire :** Représenter l'évolution de l'état de la mémoire (liste chaînée) suite à l'exécution des requêtes ci-dessus dans chacun des cas suivants :

- Le placement en mémoire centrale obéit à la stratégie First Fit.
- Le placement en mémoire centrale obéit à la stratégie Best Fit.
- Le placement en mémoire centrale obéit à la stratégie Worst Fit.

### **Exercice 2 : Mémoire virtuelle : Segmentation [4.5 Pts]**

Soit un système dont l'espace d'adressage de la mémoire virtuelle utilise la technique de la segmentation simple. Soit la table des segments représentée ci-dessous :

Segment	Base	Longueur
0	330	124
1	810	211
2	111	99
3	498	302
4	0	42
5	200	107

**Travail à faire :** Calculer l'adresse effective (physique) correspondante à chacune des adresses virtuelles (logiques) suivantes en indiquant s'il y a des adresses invalides :

1.  $\text{@}_{\text{virtuelle}} = (0 : 099)$
2.  $\text{@}_{\text{virtuelle}} = (2 : 078)$
3.  $\text{@}_{\text{virtuelle}} = (1 : 265)$
4.  $\text{@}_{\text{virtuelle}} = (3 : 222)$
5.  $\text{@}_{\text{virtuelle}} = (0 : 111)$
6.  $\text{@}_{\text{virtuelle}} = (6 : 084)$

### **Exercice 3 : Mémoire virtuelle : Pagination [5.5 Pts]**

Soit un système dont l'espace d'adressage de la mémoire virtuelle utilise la technique de la pagination. La taille d'un cadre (frame) en mémoire physique correspond à 100 mots mémoire.

**Travail à faire :**

1. Un processus possède un espace virtuel de 600 mots. Pendant l'exécution de ce processus, la suite des adresses virtuelles suivantes est appelée :

**234; 69; 123; 510; 145; 12; 456; 211; 345; 482; 14; 478; 336; 507.**

Donner la suite de numéros de pages référencées.

2. L'espace disponible en mémoire centrale pour ce processus est de 300 mots (initialement vide). Calculer le nombre de défauts de page pour les algorithmes suivants :
  - a. Optimal
  - b. Seconde chance
  - c. LFU

**% Bon travail %**

## Examen Session Principale

Niveau d'étude : L1-SI	Documents : Non autorisés
Matière : Fondement de réseaux	Nombre de pages : 2
Enseignant : Ben Othman Soufiene	Date : 08/05/2023

### Exercice 1 : (5 points)

1. A quoi servent un ETTD et un ETCD ?
2. Quelle est la différence entre la transmission série et parallèle ?
3. Énumérer 4 topologies physiques possibles pour un réseau local.
4. Quelles sont les différences entre une transmission asynchrone et une transmission synchrone ?
5. Préciser la différence entre le routage statique et le routage dynamique.
6. Citer les différents types d'adresses utilisées dans les réseaux. Justifier leur coexistence.
7. Préciser à quelle couche OSI appartient chacun les termes suivants : PPP, Routeur, Hub, UDP, RJ45, Fanion, Switch, Numéro de port.

### Exercice 2 : (5 points)

Une connexion numérique en full-duplex est établie entre 02 points A et B de la terre via un satellite géostationnaire située à 36000 km de chacun des deux points. Un signal est émis depuis A à raison de 64 Kbits/sec et ou la vitesse de propagation dans l'air est égale à 300000 km/sec. Une fois le début du signal reçu, B retourne un acquittement (réponse).

1. Calculer le temps T1 que mettra le premier bit pour arriver à B.
2. Combien de bits pourront être émis par A avant que ce dernier ne sache que B a bien reçu les premières informations.
3. Sachant que la station A désire envoyer vers B une chaîne d'informations de la taille =96Kbits. Calculer le temps T2 total de transfert de cette chaîne.

### **Exercice 3 : (5 points)**

Soit la séquence binaire suivante : 1100111101

1. Codez la séquence envoyée en bande de base par Bipolaire, Manchester, Miller.
2. Pour l'envoyer dans un réseau étendu, Codez la séquence précédente en large bande : utilisant la modulation d'amplitude 2 niveaux (10V, 20V), de fréquences 4 fréquences ( $f_1, f_2, f_3, f_4$ ), et de phase 4 phases ( $0, \pi/2, \pi, 3\pi/2$ ).

### **Exercice 4 : (5 points)**

Une entreprise a demandé et reçu l'adresse réseau **185.177.0.0**.

1. Déterminer la classe de ce réseau ?
2. Quel(s) octet(s) représente(nt) la portion réseau d'une adresse IP de cette classe ?
3. Quel(s) octet(s) représente(nt) la portion hôte d'une adresse IP de cette classe ?
4. Quel est le masque par défaut de sous-réseau ?
5. Déterminer le nombre maximal de machines que nous pouvons l'avoir dans ce réseau ?

Vous souhaitez diviser votre réseau physique en quatre sous-réseaux.

6. Déterminez le nombre de bits que vous devrez emprunter à la portion hôte de l'adresse réseau.
7. Remplir le tableau suivant :

N° de sous-réseau	Valeur binaire des bits empruntés	Plage décimale des sous-réseaux et des hôtes
Sous-réseau 1		
Sous-réseau 2		
Sous-réseau 3		
Sous-réseau 4		

8. Quel masque de sous-réseau doit être utilisé (représenter le masque de sous-réseau en décimal et en binaire) ?
9. Combien de bits reste-t-il pour affecter une adresse aux hôtes ?
10. L'adresse IP de l'hôte " A " est 185.177.7.15. L'adresse IP de l'hôte " B " est 185.177.222.97. Ces hôtes font-ils partie du même sous-réseau ? Pourquoi ?

Session de Mai 2023  
Epreuve d'Analyse II

Documents et calculatrices : Non autorisés

**Exercice 1** (6pts) Montrer que les séries numériques suivantes sont convergentes et calculer leurs sommes.

1.  $\sum_{n \geq 0} \frac{n^2}{n!}$ . Sachant que  $\sum_{n=0}^{+\infty} \frac{1}{n!} = e$ .
2.  $\sum_{n \geq 1} \frac{1}{n(n+1)}$ .
3.  $\sum_{n \geq 0} \frac{n+1}{5^n}$ .

**Exercice 2** (6pts) Déterminer, en justifiant votre réponse, le rayon de convergence des séries entières suivantes:

1.  $\sum_{n \geq 1} \ln(n) x^n$ .
2.  $\sum_{n \geq 0} \left( \frac{n+2}{2n+1} \right)^{3n} x^n$ .
3.  $\sum_{n \geq 0} \frac{(n!)^3}{(3n)!} x^{3n+1}$ .

**Exercice 3** (8pts) On se propose de développer en série entière la fonction définie par :

$$f(x) = \frac{1}{2} \arctan \left( \frac{x^2 - 1}{x^2 + 1} \right).$$

1. Donner les ensembles de définition et de dérivabilité de  $f$ .
2. Montrer que  $\forall x \in \mathbb{R}, f'(x) = \frac{x}{1+x^4}$ .
3. Sachant que  $\forall x \in ]-1, 1[, \frac{1}{1+x} = \sum_{n=0}^{+\infty} (-1)^n x^n$ , donner un développement en série entière de  $f'$  au voisinage de 0.
4. En déduire le développement de  $f$  en série entière au voisinage de 0 et préciser le rayon de convergence  $R$ .

# Examen (session principal)

Filière : LSI

Niveau : 1<sup>er</sup> année 1 et 4

Matière : Atelier de programmation II

Documents : non autorisés

## Exercice 01 :

- Si l'on suppose que les opérations suivantes sont effectuées sur une pile P contenant des chaînes de caractères, qu'est-ce qui sera affiché à l'écran par ce bout de programme ?

```
CelluleP *P=ini();
char Mot[10], Mot2[10];
P=Empiler (P, "le");
P=Empiler (P, "ferme");
P=Empiler (P, "la");
P=Depiler (P, Mot);    printf("%s \n", Mot);
P=Empiler (P, "voile");
P=Empiler (P, "belle");
P=Empiler (P, "rapidement");
P=Depiler (P, Mot);
P=Depiler (P, Mot);    printf("%s \n ", Mot);
P=Depiler (P, Mot2);  printf("%s \n", Mot2);
P=Depiler (P, Mot2);  printf("%s \n ", Mot2);
printf("%s \n", Mot);
```

- Si l'on suppose que les opérations suivantes sont effectuées sur une file F contenant des chaînes de caractères, qu'est-ce qui sera affiché à l'écran par ce bout de programme ?

```
char Mot[10], Mot2[10];
CelluleF *F=ini();
F = Enfiler (F, "le");
```

```

F = Enfiler (F, "ferme");
F = Enfiler (F, "la");
F = Defiler (F, Mot);           printf("%s \n", Mot);
F = Enfiler (F, "voile");
F = Enfiler (F, "belle");
F = Enfiler (F, "rapidement");
F = Defiler (F, Mot);
F = Defiler (F, Mot);           printf("%s \n", Mot);
F = Defiler (F, Mot);
F = Defiler (F, Mot2);          printf("%s \n", Mot2);
F = Defiler (F, Mot2);          printf("%s \n ", Mot2);
printf("%s \n", Mot);

```

### **Exercice 02 :**

Soit une liste simplement chainée dont chaque noeud est défini de la manière suivante :

```

typedef struct noeud
{
    int valeur ;
    struct noeud * suivant ;
}
noeud;

```

1. Écrire une fonction `liste_vide()` qui vérifie si la liste est vide.
2. Écrire une fonction `somme()` qui calcule la somme des éléments de la liste simplement chainée.  
Par exemple, la liste composée des entiers 25, 52, 31, 13, 20,2 aura pour somme 143.
3. Écrire une fonction `invers()` qui inverse la liste simplement chainée.  
Par exemple, la liste composée consécutivement des entiers 25, 52, 31, 13, 20,2 sera inversée de la manière suivante : 2, 20, 13, 31, 52, 25.

### **Exercice 03 :**

Nous voulons modéliser une file constituée de piles d'assiettes sales dans une cuisine de restaurant. Chaque pile d'assiettes est posée au fur et à mesure qu'elles arrivent en cuisine dans une file. Le plongeur nettoie les assiettes en les prenant, une par une, sur le dessus de la première pile stockée.

1. Proposez une structure de données qui permette de modéliser le problème.
2. Écrire une fonction permettant d'ajouter une pile d'assiettes.

3. Écrire une fonction permettant au plongeur de retirer une assiette pour la nettoyer (dans la première pile introduite).
4. Écrire une fonction permettant de copier la pile d'assiettes, c'est-à-dire créer une nouvelle pile identique à la pile initiale.
5. Écrire une fonction qui permet de trier les piles de la file de sorte que les piles contenant des assiettes paires soient placées avant les piles contenant des assiettes impaires. Cela devrait être fait tout en préservant l'ordre relatif des piles dans la file.
6. Écrire une fonction permettant de vider la pile d'assiettes, c'est-à-dire supprimer toutes les piles et les assiettes stockées dans la file.



## Matière : Programmation Python

Niveau/Section : L1/SI

Examen (Session Principale)

Date : 12 Mai 2023

Enseignante : Hajer Triki

Durée : 1h30mns

Les documents ne sont pas autorisés

L'épreuve contient 3 parties indépendantes

---

### Partie I : Questions du cours : (5 points)

1. On dit que Python est un langage caractérisé par le typage dynamique. Expliquer.
2. Quels sont les structures de données qui peuvent être les clés d'un dictionnaire ? pourquoi ?
3. Simplifier l'écriture du nombre suivant en utilisant la notation scientifique :

**9872134567793x10<sup>6</sup>**

4. Commenter les 2 codes suivants et déduire leurs résultats d'exécution :

a. 

```
def carre(x):
    y=x**2
    return y

y=5
resultat = carre(y)
print(y)
print(resultat)
```

b. 

```
def carre(x):
    global y
    y=x**2
    return y

y=5
resultat = carre(y)
print(y)
print(resultat)
```

## Partie II : Analyse d'un code Python : (5points)

Soit le programme Python suivant :

```
from tkinter import *
fen = Tk()
def action(event):
    l_fact=[]
    listeNotes=saisie_notes.get().split(",")
    for i in listeNotes:
        f = 1
        for j in range(1,int(i)+1):
            f=f*j
        l_fact.append(f)
    saisie_coeff.delete(0,END)
    saisie_coeff.insert(0,l_fact)

lbl1 = Label (text = "Enter the list of numbers:")
lbl1.place (x = 50, y = 20)
saisie_notes= Entry ()
saisie_notes.place (x = 300, y = 20)

saisie_notes.bind('<KeyPress-F>' , action)

lbl2 = Label (text = "Here are the factorial corresponding:")
lbl2.place (x = 50, y = 50)
saisie_coeff = Entry ()
saisie_coeff.place (x = 300, y = 50)

fen.mainloop ()
```

1. Délimitez les étapes de création d'une application graphique.
2. Quel est le rôle de l'instruction : `saisie_notes.bind('<KeyPress-F>',action)` ?
3. Donnez le résultat d'exécution de ce programme.

### **Partie III : Statistiques avec Python: (10 points)**

On veut réaliser des statistiques sur la distribution des salaires de N employés dans une entreprise.

On considère, dans ce cadre, que cette distribution est représentée par une liste de N salaires appelée **salary**. (N.B : dans cette distribution les salaires peuvent se répéter)

On vous demande d'écrire :

1. Une fonction **InputTheSalaries(N)** permettant de saisir et de retourner une liste de N salaires.

2. Une fonction **StatisticalSerial(L)** permettant de déterminer et de retourner la série statistique associée à la liste des salaires L. (indication : cette fonction cherche les couples des nombres d'occurrence de chaque valeur salaire et renvoie la série statistique:

$$S = [ (s_1, n_1), (s_2, n_2), \dots, (s_p, n_p) ].$$

3. Une fonction **ArithmeticAverage(S)** permettant de calculer et de retourner la **moyenne arithmétique** de la série statistique S. (indication : la moyenne arithmétique est trouvée par : somme totale des salaires/nombre des salaires)

4. Une fonction **Frequency (S)** permettant de calculer et de retourner la **liste des fréquences** de la série statistique S. (indication : La fréquence d'une salaire correspond au : nombre d'occurrence de cette salaire/nombre des salaires)

5. Une fonction **Variance(S)** permettant de calculer et de retourner la **variance** de la série statistique S. (indication : la variance se calcule par la formule  $V = (\sum (x-\mu)^2) / N$ .

où :

- x représente les si (les différentes salaires),
- $\mu$  : la moyenne arithmétique de S ,
- N : le nombre des salaires.

6. Une fonction **Ecartype (S)** permettant de calculer et de retourner l'**écart type** de la série statistique S. (indication : l'écart type se calcule par la formule suivante :  $E = \sqrt{variance}$  )

7. Un programme python permettant de créer la distribution **salary** et de tester le reste des fonctions mentionnées ci-dessus.

😊 Bon Travail 😊