

الجمهورية الجزائرية الديمقراطية الشعبية

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



ÉCOLE SUPÉRIEURE EN SCIENCES ET TECHNOLOGIES
DE L'INFORMATIQUE ET DU NUMÉRIQUE

Concours national d'accès au second cycle de l'école ESTIN

Année universitaire : 2023

Épreuve d'Informatique

Domaine : MI Durée : 2 heures Coefficient : 1

Instructions générales (à lire avant le début de l'épreuve)

- Les candidats doivent vérifier que le sujet comprend 5 pages.
- Les candidats sont invités à porter une attention particulière à la présentation.
- Les candidats doivent rendre leurs copies même vierges.
- Si au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en indiquant les raisons des initiatives qu'il est amené à prendre.
- Les numéros des questions doivent être transcrits clairement sur les copies.
- Les pages des copies et des feuilles intermédiaires doivent être numérotées.
- Les documents sont interdits, sauf indication contraire sur le sujet.
- Aucun échange n'est autorisé entre les candidats.
- Les parties *Architecture des ordinateurs II*, *ASDD*, *SFSD* et *POO* doivent être rédigées sur des copies séparées.

Barème de notation.

Partie I (*Architecture des ordinateurs II*) : 8 points.

Partie II (*ASDD*) : 4 points.

Partie III (*SFSD*) : 4 points.

Partie IV (*POO*) : 4 points.

Partie 1 : Architecture des ordinateurs II

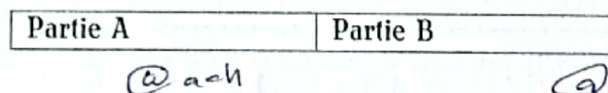
Questions de cours : (04 points)

- ✓ 1. Donner un tableau comparatif entre les mémoires statiques et les mémoires dynamiques.
- 2. Quelle est la différence entre l'écriture immédiate et l'écriture différée dans une mémoire cache?
- ✓ 3. Proposer un schéma détaillé d'un système d'interruptions hiérarchisé à 2 niveaux de priorités tels que le niveau 1 regroupe 3 causes d'interruptions et le niveau 2 regroupe 2 causes d'interruptions.
- ✓ 4. Dans le mode par accès direct en mémoire (DMA), le contrôleur de périphérique peut-il être relié à plusieurs périphériques? Justifier votre réponse. *arbitraire*

Exercice : (04 points)

Un équipement informatique d'interconnexion est doté de plusieurs ports qui sont étiquetés par **PRTN** (sachant que N est le numéro de port). Chaque port est associé à une adresse codée sur 12 bits ^(a). Cet équipement a pour rôle d'acheminer des messages à travers ces ports. Chaque message donc, incorpore une adresse codée également sur 12 bits appelée Adresse message **@msg**. Les adresses des ports sont appelées des Adresses d'acheminement **@ach** qui peuvent correspondre à une ou plusieurs adresses messages.

On suppose que l'adresse d'acheminement peut être scindée en deux parties A et B :



Dans ce contexte, une adresse d'acheminement correspond à une adresse d'un message si :

- Les bits de poids fort de la partie A de l'adresse d'acheminement sont identiques à ceux de la partie A de l'adresse message,

Dans notre contexte, on a également :

- La taille de la partie A + La taille de la partie B = 12bits
- La taille de la partie A varie de 0 à 12 bits et est appelée le masque d'acheminement.
- La taille de la partie B varie également de 0 à 12 bits.
- La partie B dans l'adresse **@ach** est constituée d'une succession de bits à 0.
- Une adresse d'acheminement peut être constituée uniquement de la partie A ou de la partie B ou les deux en même temps.

Pour effectuer cette recherche de correspondance entre l'adresse @ach et l'adresse @msg avec un masque d'acheminement, on utilise la mémoire associative.

Avant d'acheminer un message à travers l'équipement, il est donc nécessaire de rechercher dans la mémoire associative une adresse @ach qui correspond à l'adresse @msg avec le plus grand masque d'acheminement. Cette recherche permet de déterminer le port approprié pour l'acheminement du message.

Soit le contenu de la mémoire associative de l'équipement ci-dessus :

Adresse d'acheminement	Port de Sortie
110111110110	PRT1
110111110000	PRT2
110001110110	PRT4
110111000110	PRT4
100000000000	PRT3

Questions :

1. Énumérer les adresses d'acheminement qui correspondent à l'adresse message suivante : 110111110111 (sans détailler les étapes).
2. Proposer un algorithme qui permet de sélectionner le port approprié pour l'acheminement du message $(b_{12}b_{11}...b_1)$.

Partie II : ASDD

Exercice : (4 points)

Soit R la racine d'un arbre binaire de recherche d'entiers.

1. Écrire une procédure récursive AfficheDecr qui affiche les valeurs de R dans l'ordre décroissant.
2. Écrire une fonction récursive Nbr_Feuilles qui calcule le nombre de feuilles de R .
3. Écrire une fonction récursive Profondeur qui calcule la profondeur de R (sachant que le niveau de la racine est 1).

Partie III : SFSD

Exercice : (4 points)

Soit un fichier F1 de données non ordonné, formé par un ensemble de blocs contigus (TNOF), les enregistrements sont à format fixe. Nous disposons aussi d'une table d'index dense en mémoire centrale. On veut transformer F1 en un fichier F2 vu comme un tableau ordonné avec des enregistrements à taille fixe TOF.

1. Donner les déclarations du fichier de données et de la table d'index
2. Écrire un algorithme qui permet de transformer F1 en F2.

Partie IV : POO

Exercice : (4 points)

Description :

On veut mettre en œuvre une classe appelée **CustomSet** qui représente un ensemble d'étudiants.

La classe **CustomSet** doit implémenter l'interface Set.

La classe **CustomSet** doit avoir les caractéristiques suivantes :

1. Elle doit stocker des objets de la classe Student, qui possède trois attributs : ID, Nom et Prenom, Tab des notes.
2. Deux objets **Student** sont considérés comme égaux s'ils ont le même ID. *egale (obj)*
3. Les objets **Student** doivent être classés en fonction de leurs IDs. *comparable*
4. La classe **CustomSet** doit utiliser deux tableaux pour stocker les objets **Student** :
 - **principal_array** : un tableau qui stocke les objets sans redondance (ID unique).
 - **redundancy_array** : un tableau qui stocke les objets avec redondance (ID en double).
5. Le nombre maximum d'objets **Student** pouvant être stockés dans le **CustomSet** est défini par la valeur de N (N pour chaque tableau).

Travail demandé :

1. Quelle est la différence, en termes de (a) structure et (b) comportement, entre les implémentations de l'interface Set et de l'interface List?
2. Que faut-il ajouter à la classe **Student** pour compter le nombre d'objets créés?
3. Implémentez la classe **Student** avec les exigences suivantes :

- Initialisez les attributs (**ID**, **Nom** et **Prenom**).
- Ajouter la méthode avg() qui renvoie la moyenne des notes.
- Ajouter les méthodes (deux méthodes) qui examine le classement et l'égalité des objets en utilisant les structures prédéfinies de Java.

4. Implémentez la classe **CustomSet** avec les exigences suivantes :

- ✓ a. Définissez les attributs : **principal_array**, **redundancy_array**, **maximumSize** de la classe et définissez le constructeur qui prend en paramètre la taille maximale (N) de l'ensemble.
- ✓ b. Implémentez les méthodes suivantes de l'interface **Set** :
 - **boolean contains(Student e)** : Renvoie **true** si l'ensemble contient l'objet **Student** donné, **false** sinon.
 - **boolean clear()** : Supprime logiquement tous les éléments de l'ensemble et renvoie **true**.

BON TRAVAIL