Partie I
(12 points)

Structures de données statiques: (3pts)

Soit deux tableaux T1 et T2 d'entiers et de tailles respectives L1 et L2 (L1>=L2) :

Exemple :

T1 :

2	4	15	11	14	19	22	25	11	14	12	11	14	19

T2 :

11	14	19

La suite T2 se répète deux fois dans T1

Ecrire l'analyse et l'algorithme d'une fonction qui donne le nombre de fois que T2 se répète dans T1.

Structures de données dynamiques : (3pts)

Soit un arbre binaire d'entiers.

1. Ecrire une fonction récursive qui détermine si l'arbre binaire est pur ou pas. Un arbre binaire est pur si chaque nœud est une feuille ou a exactement deux fils.

2. Ecrire une fonction récursive de type entier qui transforme un arbre binaire en un arbre binaire pur et qui donne le nombre de zéros rajoutés.

Structures de fichiers : (3pts)

Considérons la méthode: -essai linéaire en mémoire centrale, pour cette méthode on prendra la table **Tab1 [0..6]** avec la fonction de hachage **h (x) = x mod 7,**

- Donnez l'algorithme de recherche/insertion -Essai Linéaire (cas général).
- Considérez que chaque case du tableau peut contenir 2 éléments. Remplissez la table Tab1 avec les données suivantes **22, 0, 45, 1, 8, 3, 10, 6, 13, 14, 21, 5**

Programmation Orientée Objet : (3pts)

Étant donné l'énoncé suivant :

Une société de prestation offre plusieurs services qui sont représentés par la classe **Service**.

- Elle est caractérisée par un identificateur, un nom et un coût.
- Les attributs de la classe **Service** sont accessibles partout dans le programme.
- Elle possède un constructeur exhaustif (qui initialise tous les attributs).
- Deux objets de cette classe sont égaux si leurs identificateurs sont égaux.
- L'affichage de tout objet ***ser*** de Type **Service** (avec l'instruction *System.out.println(ser)*) devra afficher le nom et le coût de cet objet.

1. Donnez le code java de la classe **Service** qui répond à cet énoncé. **(1 pt)**

Soit l'interface **Critere** définie ainsi :

```
interface Critere {
    boolean estSatisfaitPar(Service s);
}
```

Où la méthode ***estSatisfaitPar(Service s)*** doit retourner *true* si et seulement si le service s satisfait le critère.

2. Donnez le code de la classe **CritereCout** qui implémente l'interface **Critere** et dont le critère est satisfait lorsque le coût du service est inférieur à un coût donné lors de l'instanciation. **(1 pt)**

La classe **Societe** contient une liste de services ***listeServices*** de type **ArrayList<Service>**, ainsi qu'une méthode ***SelectionServices(CritereCout c)*** qui retourne parmi tous les services contenus dans ***listeServices*** ceux qui satisfont le critère de coût donné en paramètre.

3. Donnez le code java de la méthode *SelectionServices(CritereCout c)* **(1 pt)**

Partie II : Architecture des ordinateurs
(8 points)

Exercice 1: (2 points)

On veut réaliser un système d'alarme qui permet de surveiller quatre locaux avec des urgences différentes.
Des capteurs placés dans chaque local permettent d'envoyer un signal (L0, L1, L2, L3) en cas d'alerte, selon le local.

Ordre de priorité : L0 > L1 > L2 > L3

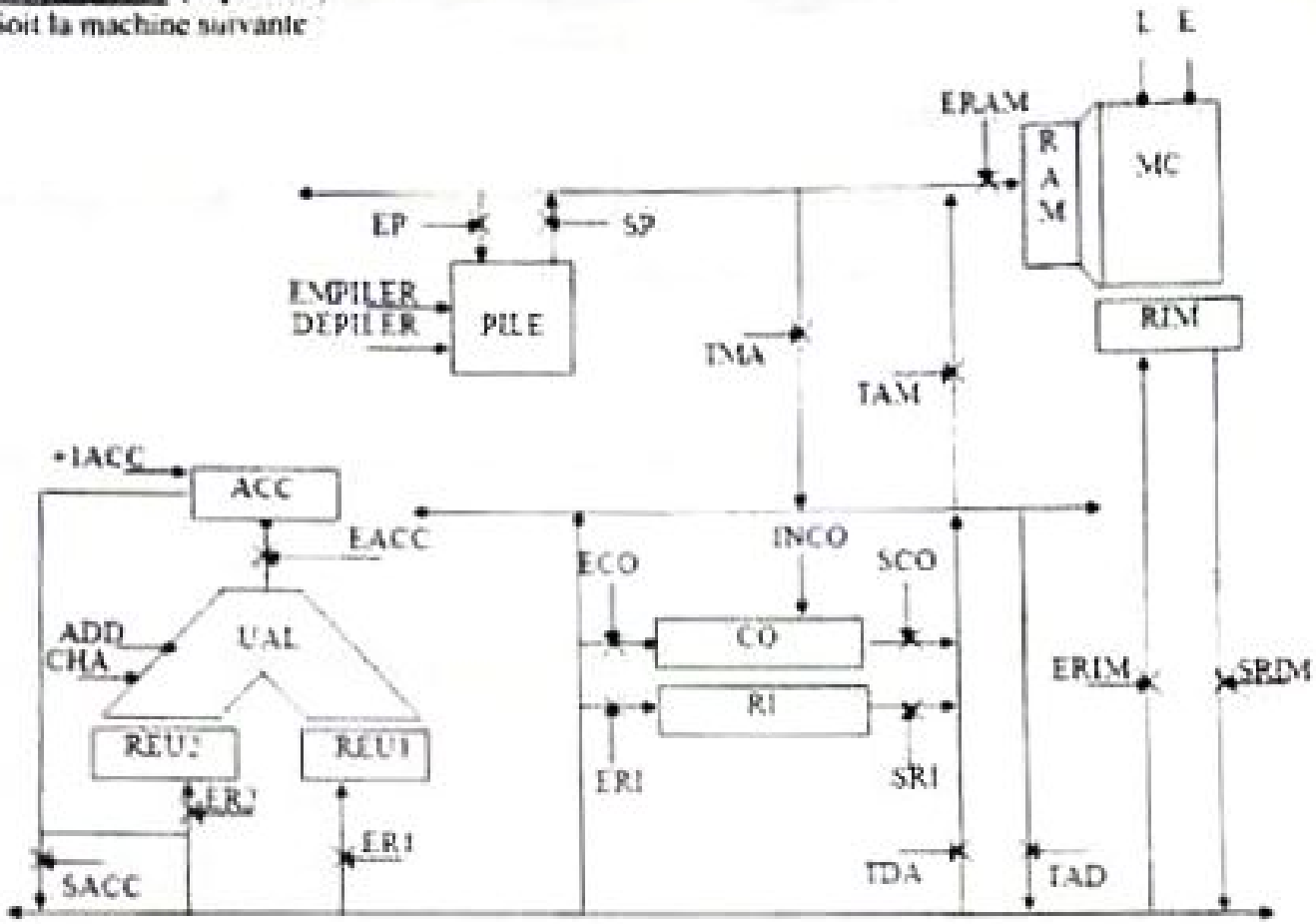
Le circuit génère en sortie le numéro du local le plus prioritaires parmi ceux qui ont généré un signal d'alerte.

Question :

1- Faire la table de vérité du circuit.
2- Réaliser le schéma du circuit avec une ROM.

Exercice 2: (3 points)

Soit la machine suivante



La commande CHA permet de charger l'accumulateur avec le contenu du registre REU1.

Questions :

1. Donner les micro-instructions et microcommandes de l'instruction de sauvegarde du contenu de l'accumulateur dans la pile
2. Donner les micro-instructions et microcommandes de l'instruction de restauration du contenu de l'accumulateur depuis la pile

Exercice 3 : (3 points)

- On veut réaliser un système d'entrées/sorties sur disque.
- Ce système est composé de l'unité centrale, de la mémoire centrale, d'un contrôleur de type DMA et d'un disque dur.
- Le contrôleur est capable d'exécuter des commandes d'Entrées/Sorties complètes initialisées par l'unité centrale et d'accéder seul à la mémoire centrale. Pour cela, il communique avec l'unité centrale, la mémoire centrale et le périphérique.
- Pour régler les éventuels conflits d'accès au bus et à la mémoire centrale, on dispose d'un **arbitre** qui donne la priorité **maximale à l'unité centrale**.

Questions :

1. Donner la liste des registres du contrôleur en détaillant leur nom.
2. Donner sur un schéma les connexions du contrôleur avec le disque, l'unité centrale, l'arbitre de priorité et la mémoire centrale en représentant les bus nécessaires.
   Montrer sur le même schéma les différentes ***étapes de dialogue*** entre l'UC, le contrôleur, le disque, l'arbitre et la mémoire centrale, lorsqu'une opération de lecture est lancée par l'Unité Centrale, pour un nombre ***Nbre*** d'octets, depuis le disque (***piste x, secteur y***) vers la mémoire centrale (***adresse ADR***). Associer à chaque étape un numéro sur le schéma.
3. Expliquer dans un tableau chaque étape représentée dans le schéma en donnant le numéro et le rôle de chaque étape.