

**Suret  de Fonctionnement -TD 1- Entraves de la Suret  de Fonctionnement**

**Exercice 1\* :**

On consid re un distributeur de boissons chaudes qui permet de d livrer des boissons et rendre la monnaie

**1. Classer ces observations :**

- la s lection de Makoccino n'est plus possible pour tous les utilisateurs
- le distributeur est trop lent
- le distributeur ne rend pas la monnaie   personne
- le distributeur est en panne puis hier
- Certain utilisateurs consid re qu'Expresso est un peu l ger.
- La distribution de th  glac  n'est pas possible entre 8h et 9h.
- Les pi ces de 50DA ne sont plus accept es.
- La monnaie n'est pas rendue.

**2. Classer ces Fautes ou Erreurs :**

- Ressource  puis e (Goblet) sans message.
- Bouton pour Verveine est obstru 
- Un lien de communication perd le message
- Un c ble est d connect 
- Le c ble est us .



**Exercice 2**

Quelle est le Comportement attendu des syst mes suivants ? Citer des exemples de d faillances ?

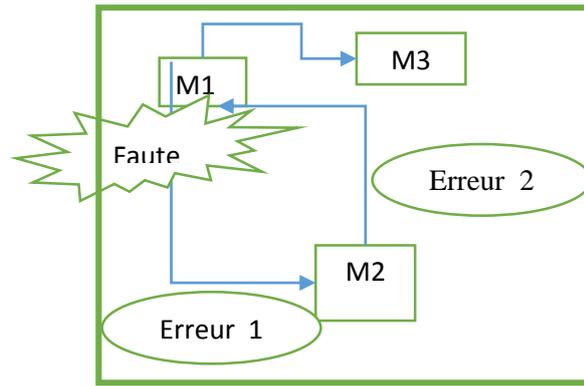
- ampoule
- imprimante



**Exercice 3:**

Soit le syst me structur  de 3 machines suivant:

- Repr senter les niveaux de la contamination des erreurs et de la d faillance.
- Effectuer des calculs de latence par somme de latences locales : faute   l'instant 0 ; latence L1 dans M1, propagation   travers M2, M3 latence L2 et L 3 et enfin affectation de la sortie.



L1=10ms, L2=100ms, L3=30ms.

**Exercice 4\* :**

Soit un compteur binaire asynchrone par 16 qui compte des impulsions arrivant sur l'entr e E et affiche le r sultat sur S en code binaire naturel sur 4 bits a,b,c,d. Une d faillance durable de collage   0 de la sortie de poids a se produit alors que le compteur est dans l' tat initiale 0000. On  volue donc avec des Configurations 0,b,c,d.

Calculer la latence moyenne de cette faute sachant que l'intervalle de temps moyen entre deux  v nements d'entr e est de 2 millisecondes.

**Exercice 5\*:**

Soit le programme erron  ci apr s. Pour x = [-4,2,0,2], ce programme provoque une d faillance (r sultat attendu : 2).

- Identifiez la faute.
- Si possible, identifiez un cas de test qui n'ex cute pas la faute.
- Si possible, identifiez un cas de test qui ex cute la faute, mais n'aboutit pas   un  tat d'erreur. (d) Si possible, identiez un cas de test qui entra ne une erreur, mais pas une d faillance.
- Pour le cas de test donn , identifiez le premier  tat d'erreur et d crivez-le
- Corrigez la faute et v rifiez que le test donn  produit maintenant le r sultat attendu.

```

// Effects: if x==null throw NullPointerException
// else return the number of positive elements in x
public static int countPositive (int [] x) {
    int count = 0;
    for (int i=0; i < x.length; i++) {
        if (x[i] >= 0) {
            count++;
        }
    }
    return count;
}
}

```

