


Institut Supérieur d'Informatique et de Mathématiques  	Année Universitaire : 2024-2025
	<b>Examen</b>  Matière : ASD 1 Filière : L1 Info Enseignant : Taoufik Sakka Rouis

### Exercice 1 : (6+7+7 Points)

On considère qu'un tableau est dit trié de manière hybride s'il respecte les règles suivantes :

- Tous les éléments **pairs** doivent être situés dans la partie gauche et triés en ordre **croissant**.
- Tous les éléments **impairs** doivent être situés dans la partie droite et triés en ordre **décroissant**.

**Exemple :** 2 | 4 | 18 | 77 | 33 | 21 | 7

Pour cela :

1. En se basant sur le principe du **tri par insertion** (qui suppose que les  $i-1$  premiers éléments sont déjà triés et consiste à insérer l'élément situé à la position  $i$  à sa place appropriée parmi ces  $i-1$  éléments), implémentez une fonction qui trie tous les éléments du tableau en ordre **croissant**.

⇒ Cette fonction doit avoir le prototype suivant : **void TriInsertion(int T[], int n)**

2. Implémentez une seconde fonction qui prend en paramètre un tableau **T trié** et le parcourt élément par élément. Pour chaque élément **impair**, elle l'insère dans la partie droite du tableau de manière à maintenir cette partie triée en ordre **décroissant**.

⇒ Cette fonction doit avoir le prototype suivant : **void TriHybride(int T[], int n)**

3. Implémentez une troisième fonction qui effectue une recherche **dichotomique adaptée aux tableaux triés de manière hybride**. La fonction doit retourner la position de l'élément  $x$  dans le tableau s'il est trouvé ; -1 sinon.

⇒ Cette fonction doit avoir le prototype suivant : **int RechDichHybride(int T[], int n, int x)**

**NB.**

- La note sera attribuée en fonction de la complexité de votre solution.
- Vous ne devez pas utiliser de tableau supplémentaire.