

Nom: Andy Tran

## Directives

Ce travail est **individuel**. Vous devez répondre directement dans le fichier.

## Modalités de remise

Le travail doit être remis à la date indiquée sur **Léa**. Il doit être remis en format **.pdf**. (Si vous ne savez pas comment faire, demandez à votre enseignant).

### Question 1 (8 points)

Soit la méthode `ecrireFichier()` suivante qui sauvegarde des données dans un fichier binaire.

```
private static void ecrireFichier() {  
    try (DataOutputStream ecrivain = new DataOutputStream(new  
        FileOutputStream("mystere.data"))) {  
        ecrivain.writeUTF("Bonjour mon ami !!");  
        ecrivain.writeInt(3);  
        ecrivain.writeUTF("C'est enfin terminé...");  
        ecrivain.writeBoolean(true);  
    } catch (FileNotFoundException e) {  
        System.out.println("Fichier inexistant.");  
    } catch (IOException e) {  
        System.out.println("Erreur d'accès au fichier.");  
    }  
}
```

Complétez la méthode `ecrireMessages()` suivante afin qu'elle lise les données du fichier et affiche les chaînes suivantes :

```
Bonjour mon ami !!  
Bonjour mon ami !!  
Bonjour mon ami !!  
C'est enfin terminé...
```

Vous devez déclarer et lire les variables à l'intérieur du fichier binaire produit par la méthode `ecrireFichier()`. Écrivez votre code dans la zone blanche.

```
private static void ecrireMessage() {  
    try (DataInputStream lecteur = new DataInputStream(new  
        FileInputStream("mystere.data"))) {  
        String messageSalutation = lecteur.readUTF();  
        int nombre = lecteur.readInt();  
        String messageFin = lecteur.readUTF();  
        Boolean salutationNecessaire = lecteur.readBoolean();  
  
        for (int i=0; i<nombre; i++) {  
            if (salutationNecessaire) {  
                System.out.println(messageSalutation);  
            }  
        }  
        System.out.println(messageFin);  
  
    } catch (FileNotFoundException e) {  
        System.out.println("Fichier inexistant.");  
    } catch (IOException e) {  
        System.out.println("Erreur d'accès au fichier.");  
    }  
}
```

## **Question 2 (8 points)**

Écrivez une méthode qui reçoit en paramètre une `ArrayList<Integer>`. Vous devez inverser tous les éléments de la liste en utilisant une instance de `Stack<Integer>`. Pour ajouter et retirer des éléments de la pile, utilisez les méthodes `push()` et `pop()`.

```
private static void inverserListe(ArrayList<Integer> laListe) {  
  
    Stack<Integer> outil = new Stack<Integer>();  
    for(int i = 0; i<laListe.size(); i++){  
        outil.push(laListe.get(i));  
    }  
    laListe.clear();  
    int taille = outil.size();  
    for(int i = 0; i < taille; i++){  
        laListe.add(outil.pop());  
    }  
  
}
```

**Question 3 (3 points)**

Créez une classe se nommant `ExceptionValeurNegative`. Cette classe n'aura aucun comportement particulier. Elle se contentera d'hériter de `RuntimeException`.

```
public class ExceptionValeurNegative extends RuntimeException{  
    public ExceptionValeurNegative(String s){  
        super(s);  
    }  
}
```

#### **Question 4 (8 points)**

Écrivez une méthode qui reçoit en paramètre une `ArrayList<Integer>`. Cette méthode doit faire la somme de toutes les valeurs paires non négatives de la liste. Si une valeur négative est trouvée, elle doit lancer l'exception `ExceptionValeurNegative` que vous avez créée au numéro précédent.

```
private static int additionnerPairesPositives(ArrayList<Integer> list) throws
ExceptionValeurNegative {
    int somme = 0;
    for (int i = 0; i < list.size(); i++) {
        if(list.get(i) % 2 == 0){
            somme+=list.get(i);
        }else if(list.get(i)<0){
            throw new ExceptionValeurNegative("Il y a un nombre negatif");
        }
    }
    return somme;
}
```

Pour les 5 prochaines questions, une séquence aléatoire de 15 nombres entre 0 et 99 inclusivement vous a été envoyée par MIO. Cette séquence sera utilisée afin de faire les traces de tri et de recherche des prochaines questions :

Écrivez ici votre séquence aléatoire :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
84	61	24	46	52	40	38	74	72	11	85	91	25	60	88

À la première ligne de chaque question, réécrivez votre séquence aléatoire. Les tris se font en ordre croissant.

### Question 5 (4 points)

Faites la trace complète du tri **insertion**. L'élément qui a été *inséré* à chaque passe doit être en **rouge**.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
84	61	24	46	52	40	38	74	72	11	85	91	25	60	88
61	84	24	46	52	40	38	74	72	11	85	91	25	60	88
24	61	84	46	52	40	38	74	72	11	85	91	25	60	88
24	46	61	84	52	40	38	74	72	11	85	91	25	60	88
24	46	52	61	84	40	38	74	72	11	85	91	25	60	88
24	40	46	52	61	84	38	74	72	11	85	91	25	60	88
24	38	40	46	52	61	84	74	72	11	85	91	25	60	88
24	38	40	46	52	61	74	84	72	11	85	91	25	60	88
24	38	40	46	52	61	72	74	84	11	85	91	25	60	88
11	24	38	40	46	52	61	72	74	84	85	91	25	60	88
11	24	38	40	46	52	61	72	74	84	85	91	25	60	88
11	24	25	38	40	46	52	61	72	74	84	85	91	60	88
11	24	25	38	40	46	52	60	61	72	74	84	85	91	88
11	24	25	38	40	46	52	60	61	72	74	84	85	88	91

### Question 6 (4 points)

Faites la trace complète du tri **sélection**. L'élément qui a été *sélectionné* à chaque passe doit être en **rouge**.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
84	61	24	46	52	40	38	74	72	11	85	91	25	60	88
11	61	24	46	52	40	38	74	72	84	85	91	25	60	88
11	24	61	46	52	40	38	74	72	84	85	91	25	60	88
11	24	25	46	52	40	38	74	72	84	85	91	61	60	88
11	24	25	38	52	40	46	74	72	84	85	91	61	60	88
11	24	25	38	40	52	46	74	72	84	85	91	61	60	88
11	24	25	38	40	46	52	74	72	84	85	91	61	60	88
11	24	25	38	40	46	52	60	72	84	85	91	61	74	88
11	24	25	38	40	46	52	60	61	84	85	91	72	74	88
11	24	25	38	40	46	52	60	61	72	85	91	84	74	88
11	24	25	38	40	46	52	60	61	72	74	91	84	85	88
11	24	25	38	40	46	52	60	61	72	74	84	91	85	88
11	24	25	38	40	46	52	60	61	72	74	84	85	91	88
11	24	25	38	40	46	52	60	61	72	74	84	85	88	91

### Question 7 (4 points)

Faites la trace complète du tri **bulle**. Après chaque passe, l'élément qui se retrouve *bien placé* doit être en **rouge**.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
84	61	24	46	52	40	38	74	72	11	85	91	25	60	88
61	24	46	52	40	38	74	72	11	84	85	25	60	88	91
24	46	52	40	38	61	72	11	74	84	25	60	85	88	91
24	46	40	38	52	61	11	72	74	25	60	84	85	88	91
24	40	38	46	52	11	61	72	25	60	74	84	85	88	91
24	38	40	46	11	52	61	25	60	72	74	84	85	88	91
24	38	40	11	46	52	25	60	61	72	74	84	85	88	91
24	38	11	40	46	25	52	60	61	72	74	84	85	88	91
24	11	38	40	25	46	52	60	61	72	74	84	85	88	91
24	11	38	40	25	46	52	60	61	72	74	84	85	88	91
11	24	38	25	40	46	52	60	61	72	74	84	85	88	91
11	24	25	38	40	46	52	60	61	72	74	84	85	88	91
11	24	25	38	40	46	52	60	61	72	74	84	85	88	91
11	24	25	38	40	46	52	60	61	72	74	84	85	88	91
11	24	25	38	40	46	52	60	61	72	74	84	85	88	91



### Question 8 (5 points)

Faites la trace complète du tri **rapide** (sélectionnez votre pivot **au début**). Une passe est complétée après chaque partage autour du pivot. Le pivot utilisé à chaque passe doit être en **rouge**.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
84	61	24	46	52	40	38	74	72	11	85	91	25	60	88
84	61	24	46	52	40	38	74	72	11	85	91	25	88	60
25	61	24	46	52	40	38	74	72	11	85	91	84	88	60
25	11	24	46	52	40	38	74	72	61	85	91	84	88	60
25	11	24	46	52	40	38	60	72	61	85	91	84	88	74
25	11	24	46	52	40	38	60	72	61	74	91	84	88	85
25	11	24	46	52	40	38	60	72	61	74	84	91	88	85
25	11	24	46	52	40	38	60	72	61	74	84	85	88	91
25	11	24	46	52	40	38	60	73	61	84	74	85	88	91
25	11	24	46	52	40	38	60	61	73	84	74	85	88	91
25	11	24	38	52	40	46	60	61	73	74	84	85	88	91
25	11	24	38	40	52	46	60	61	73	74	84	85	88	91
25	11	24	38	40	46	52	60	61	73	74	84	85	88	91
24	11	25	38	40	46	52	60	61	73	74	84	85	88	91
11	24	25	38	40	46	52	60	61	73	74	84	85	88	91

**Question 9 (3 points)**

Écrivez ici vos valeurs en ordre croissant :

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
24	11	25	38	40	46	52	60	61	73	74	84	85	88	91

Indiquez avec quelles valeurs (et non les indices) du tableau vous allez devoir comparer les valeurs recherchées suivantes dans le cadre d'une recherche binaire:

1. -2

60, 38, 11, 24

2. 100

60, 84, 88, 91