PASSAGE DE PARAMÈTRES

Exercice 1 Soit le code Java suivant :

```
class A {
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int b=s.nextInt();
        System.out.println(a+b);
    }
}

class B {
    static int somme(int a,int b){
        return a+b;
    }
}

class C {
    static int somme(){
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int b=s.nextInt();
        return a+b;
    }
}
```

Indiquer quelles sont les instructions qui correspondent à des appels de fonction, avec quels paramètres, quelle valeur de retour, quelles lignes correspondent à la définition d'une fonction, quels paramètres sont déclarés, quel type de retour... Y a-t-il une différence entre les deux fonctions somme ?

Exercice 2 Expliquer ce que produit l'exécution du programme suivant :

```
class Z {
    static void ajouterInt(int x){
        x=x+1;
    }
    static void ajouterTab(int[] tab){
        tab[0]=tab[0]+1;
    }
    public static void main(String[] ppp){
        int x=10;
        System.out.println(x);
        ajouterInt(x);
        System.out.println(x);
        int[] t=new int[3];
        t[0]=10;
        System.out.println(t[0]);
        ajouterTab(t);
        System.out.println(t[0]);
    }
}
```

Est-ce différent si on écrit :

```
static int ajouterInt(int x){
    x=x+1;
    return x;
}
```

Doit-on changer autre chose, pour que l'effet voulu soit réalisé ? Faire un schéma du contenu de la mémoire lorsque le point d'exécution est dans la fonction `ajouterInt`, dans la fonction `ajouterTab`.

Exercice 3 Écrire une méthode statique qui reçoit un tableau d'entiers en argument, et renvoie un (autre) tableau où tous les éléments ont été incrémentés d'une unité, sans modifier le tableau initial. Écrire une ligne de programme faisant appel à cette méthode.