

<i>Terminaison et correction</i>

1. On considère la méthode “mystere” de la classe “MesEntiers” :

```
public class MesEntiers {
    /** antécédent : .....
    * conséquent : ..... */
    public static int mystere(int x) {
        int result = 0;
        if (x==0)
            result = 1;
        // A0 : .....(propriété sur x et result pour vérifier que c'est correct)
        else {
            int n = x;
            while (n!=0) {
                // A1 : .....(propriété sur n pour vérifier que ça termine)
                // A2 : .....(propriété sur result, n et x pour vérifier que c'est correct)
                n = n/10;
                result++;
            }
            // A3 : .....(propriété sur x et result pour vérifier que c'est correct)
        };
        return result;
    }
}

// le main pour tester
public static void main(String args[]){
    System.out.println("mystere de 9321 : " + mystere(9321));
}
```

- Que vaut *mystere*(9321) ?
- Donner les assertions *A0*, *A1*, *A2* et *A3* et en déduire les conséquents (antécédent et conséquent) de la procédure mystère.

2. Complétez la méthode suivante qui détermine si c est un palindrome. Un palindrome est un mot qui se lit aussi bien de gauche à droite que de droite à gauche. Par exemple, “rever” et “ressasser” sont des palindromes, “carotte” n’est pas un palindrome.

Pour accéder aux caractères d’une chaîne vous pouvez utiliser la méthode “charAt(int i)” de la classe String. Exemple : String s = new String(“carotte”); s.charAt(3) renvoie ‘o’.

Attention : à chaque étape de la boucle, vous devez respecter l’assertion A1

```
public boolean palindrome(String c) {
    int i = 0;
    int j = c.length()-1;
    while ((i<j) && .....)) {
        // A1 : si C1 est la sous-chaîne de c allant de l’indice 0 à l’indice i-1
        //      : si C2 est la sous-chaîne de c allant de l’indice j+1 à l’indice c.length()-1
        //      : alors C1C2 est un palindrome
        .....
    }
    return .....;
}
```

3. On a modifié la ligne 9 de la méthode “rechercheVite”. Trouver un exemple de tableau pour lequel cette nouvelle méthode ne termine jamais.

```
1 public static int rechercheVite(int[] tab, int x) {
2     int gauche = 0;
3     int droite = tab.length - 1;
4     int milieu ;
5     while (gauche <= droite) {
6         milieu = (gauche + droite) / 2 ;
7         if (x==tab[milieu]) return milieu;
8         if (x<tab[milieu])
9             droite = milieu; // ERREUR (dans le cours : droite = milieu - 1)
10            else gauche = milieu + 1;
11    }
12    return -1;
13 }
```

Justifiez la terminaison de la méthode “rechercheVite()” du cours.

4. Ecrire la méthode “public static int recherche(int[] tab, int x, int o)” de la classe “TableauEntier” qui renvoie l’indice de l’occurrence n° o de x ou -1 si x apparaît moins de o fois dans le tableau. Par exemple, si $tab = \{1, 2, 4, 12, 0, 1, 5\}$, recherche(tab,1,2) renvoie 5, recherche(tab,1,1) renvoie 0 et recherche(tab,1,6) renvoie -1.