Solutions

Solution 1.

```
public class TableauDEntier {

  public static boolean appartient(int[] t, int a) {
    int i;
    for (i = 0; i < t.length; i++)
      if (t[i] == a)
        return true;
    return false;
  }

  public static double moyenne(int[] t) {
    int i;
    double somme = 0;
    for ( i = 0; i < t.length; i++)
      somme += t[i];
    return somme / t.length;
  }

  // version 1
  //  public static boolean egal(int [] t1, int [] t2) {
  //      int i;
  //      if (t1.length != t2.length)
  //        return false;
  //      for ( i = 0; i < t1.length; i++)
  //        if (t1[i] != t2[i])
  //          return false;
  //      return true;
  //    }

  // version 2
  // public static boolean egal(int [] t1, int [] t2) {
  //    int i;
  //    for ( i = 0; i < t1.length && i < t2.length; i++)
  //      if (t1[i] != t2[i])
  //          return false;
  //    return (i == t1.length && i == t2.length);
  //    }

  // version 3
  public static boolean egal(int[] t1, int [] t2) {
    int i;
    if (t1.length != t2.length)
```

```
      return false;
    for ( i = 0; i < t1.length && t1[i] == t2[i ]; i++)
      ;
    return (i == t1.length);
  }

  public static void main(String[] args) {
    int [] test = {1, 2, 3};
    int [] test1 = {1, 2, 5};
    int [] test2 = {1, 2, 3};
    System.out.println("2 ? " + appartient(test , 2));
    System.out.println("5 ? " + appartient(test , 5));
    System.out.println("Moyenne ? " + moyenne(test));
    System.out.println("Egal test et test1 ? " + egal(test , test1 ));
    System.out.println("Egal test et test2 ? " + egal(test , test2 ));
  }

}
```

Solution 2.

```
public class Multiplication {

  // version 1
  //  public static int [][] tableMulti() {
  //      int ligne , colonne;
  //      int [][] t = new int [10][10];
  //      for (ligne = 0; ligne < 10; ligne++)
  //        for (colonne = 0; colonne < 10; colonne++)
  //          t[ligne ][colonne] = (ligne + 1) * (colonne + 1);
  //      return t;
  //    }

   // version 2
   // on utilise le fait que (i * j) = (j * i) pour
   // diminuer le nombre de multiplications é effectues
  public static int [][] tableMulti() {
    int ligne , colonne;
    int [][] t = new int[10][10];
    for(ligne = 0; ligne < 10; ligne++)
      for(colonne = 0; colonne <= ligne; colonne++)
        t[ligne ][colonne] = t[colonne][ligne ] = (ligne + 1) * (colonne + 1);
    return t;
  }

  public static void afficheTable(int [][] t) {
    int i , j;
```

```
    for(i = 0; i < t.length; i++) {
      for( j = 0; j < t[i].length; j++)
        System.out.print(t[i][j] + "\t");
      System.out.println();
    }
  }

  public static void main(String[] args) {
    afficheTable(tableMulti());
  }

}
```

Solution 3.

```
public class Chaine {

  public static boolean egal(String s1, String s2) {
    int i;
    int lg = s1.length();
    if (lg != s2.length())
      return false;
    for (i = 0; i < lg && s1.charAt(i) == s2.charAt(i); i++)
      ;
    return (i == lg);
  }

  public static int occurence(String s, char c) {
    int lg = s.length();
    int i;
    for (i = 0; i < lg; i++)
      if (s.charAt(i) == c)
        return i;
    return -1;
  }

 public static String extraire(String s, int debut, int fin) {
   String resultat = "";
   int i;
   for (i = debut; i < fin; i++)
     resultat += s.charAt(i);
   return resultat;
 }

  public static int nombreOccurences(char l, String mot) {
    int lg = mot.length();
    int i, cpt = 0;
```

```
  for (i = 0; i < lg; i++)
    if (mot.charAt(i) == l)
      cpt++;
  return cpt;
}

public static boolean estSousMot(String u, String v) {
  int lgu = u.length();
  int lgv = v.length();

  if (lgu > lgv)
    return false;

  int iv, iu;

  for(iu = 0, iv = 0; iu < lgu && iv < lgv; iv++)
    if (u.charAt(iu) == v.charAt(iv))
      iu++;
  return iu == lgu;
}

public static boolean estSuffixe(String u, String v) {
  int lgu = u.length();
  int lgv = v.length();

  if (lgu > lgv)
    return false;

  int iv, iu;

  for(iu = lgu - 1, iv = lgv -1; iu >= 0; iu--, iv--)
    if (u.charAt(iu) != v.charAt(iv))
      return false;
  return true;
}

public static boolean estFacteur(String u, String v) {
  int lgu = u.length();
  int l = v.length() - lgu;

  if(l < 0)
    return false;

  int iu, iv;
  for(iv = 0; iv <= l; iv++) {
```

```
      iu = 0;
      while (u.charAt(iu) == v.charAt(iv + iu)) {
        iu++;
        if (iu == lgu)
          return true;
      }
    }
    return false;
  }

  public static void main(String[] args) {
    String test = "toto";
    String test1 = "totos";
    System.out.println("test = test1 ? " + egal(test , test1 ));
    System.out.println("test = test ? " + egal(test , test ));
    System.out.println("s dans toto ? " + occurence(test, 's' ));
    System.out.println("extraire de à 1 3 dans toto ? " + extraire(test , 1, 3));
    System.out.println("t dans toto ? " + nombreOccurences('t', test));
    System.out.println("aie dans automobile ? " + estSousMot("aie", "automobile"));
    System.out.println("bile dans automobile ? " + estSuffixe("bile" , "automobile"));
    System.out.println("mob dans automobile ? " + estFacteur("mob", "automobile"));

  }
}
```