# TD2

## Exercice – graphe de flot de contrôle + couverture de code

1. Soit le programme suivant :

```
void foo(int y) {
  Scanner sc = new Scanner(System.in);
  int x = 0;
  while (y > 100){
    x = x + y;
    y = sc.nextInt();
  }
  if (y < 200) {
    System.out.println(x);
  }
  System.out.println(y);
}
```

   (a) Construisez le graphe de flot de contrôle de ce programme.

   (b) Donnez la complexité cyclomatique du graphe.

   (c) Si possible, proposez un ensemble minimal de données de test pour couvrir tous les nœuds du graphe.

   (d) Si possible, proposez un ensemble minimal de données de test pour couvrir tous les arcs du graphe.

   (e) Si possible, proposez un ensemble minimal de données de test pour couvrir tous les chemins élémentaires du programme.

2. Soit la fonction suivante dont la spécification annonce qu'elle permet de calculer le plus petit commun multiple (ppcm) de deux entiers non nuls donnés :

```
// x > 0, y > 0
int ppcm(int x, int y) {
  if (x == y)
    return x;
  int p = x * y;
  int res = 0;
  while ((p > x) && (p > y)) {
    if ((p % x == 0) && (p % y == 0)) {
      res = p;
    }
    p = p - 1;
  }
  return res;
}
```

(a) Construisez le graphe de flot de contrôle de cette fonction, en décomposant les conditionnelles.

(b) Donnez la complexité cyclomatique du graphe.

(c) Listez les sommets et les arcs parcourus par la donnée de test (x=2, y=3).

(d) Si possible, proposez un ensemble minimal de cas de test pour couvrir tous les nœuds du graphe.

(e) Si possible, proposez un ensemble minimal de cas de test pour couvrir tous les arcs du graphe.

(f) Si possible, proposez un ensemble minimal de cas de test pour couvrir tous les chemins élémentaires du programme.

(g) À l'aide de ces cas de test, trouvez le *bug* dans le code et le corriger.

## Exercice – JUnit + EclEmma

1. Implémentez les cas de test pour chacun des quatre programmes erronés du premier exercice du TD précédent en utilisant le *framework* JUnit 3.8.
2. Idem en utilisant le *framework* JUnit 4.
3. Vérifiez la couverture de vos tests (*i.e.*, atteindre 100% de couverture de code).