TD de Test Logiciel

Introduction

Exercice 1 (Petites questions)**.**

1. *On dit souvent que le test exhaustif est impossible. Que serait ce test exhaustif ? Pourquoi serait-ce impossible ?*
2. *Un défaut du test aléatoire est qu'il peine à produire certaines DT très particulières. Par exemple, quelle est la probabilité de produire (x,y) avec x=y ? (machine 32 bits)*
3. *Dites pourquoi un logiciel qui a passé avec succès tous les tests système peut échouer sur les tests d'acceptation ?*
4. *Citer un cas où un logiciel peut être perçu comme défectueux alors qu'il n'y a pas d'erreur ni de défaut.*
5. *Pourquoi est-ce que les tests réalisés pour le test unitaire pourraient ne pas être réutilisables pour le test système ?*
6. *Considérons un programme avec un petit nombre de chemins, et un jeu de test passant par tous les chemins. Les tests passent sans problème. Le programme peut-il contenir des erreurs ?*

Exercice 2 (Triangle)**.** *Un programme prend en entrée trois flottants, interprétés comme les longueurs des côtés d'un triangle. Le programme répond s'il s'agit d'un triangle scalène, isocèle ou équilatéral.*
Produire une suite de tests pour ce programme : CT et oracle.
Quel famille de sélection de test avez-vous suivie ?
Avez-vous fait seulement du test nominal ou aussi de robustesse ?
On suppose que l'en-tête de la fonction est : `kind my-prog(float a, float b, float c)`*, où* `kind` *est un type énuméré pouvant valoir SCAL, ISOC, EQUIL. Écrire un script de test pour vos cas de test.*

Exercice 3. *Soit le programme suivant :*

```
1  void myfunction(int a, int b, int c){
2  if (b < c) {
3     d = 2*b; f = 3*c;
4     if (x >= 0) {
5        y = x;   e = c;
6        if (y==0) {
7           a = f-e;
8           if (d < a)
9              write(a);
10 } } }
11 }
```

– *Faites du test BB pour déduire des DT passant par tous les chemins du programme.*
– *Déterminer l'oracle et écrire le script de test.*

Exercice 4 (eXtrem Programming)**.** *Nous allons nous renseigner un peu sur l'eXtrem Programming (XP). Essayez de trouver les infos suivantes (le web est un bon informateur) :*

1. *Que sont les méthodes agiles en général ? Pourquoi ont-elles été développées ? Citez en trois.*
2. *Qui a créée l'eXtrem Programming ?*
3. *Expliquez les termes suivants (dans le cadre de XP) : pair-programming, egoless programming, keep it simple, constant refactoring, test first, test-driven development, sustainable development, implication du client*
4. *Expliquez comment les bonnes pratiques suivantes ont été intégrées dans XP : use-cases, tests unitaires, tests de régression, développement itératif, revues, implication du client, séparation du développeur et du testeur ?*
5. *Quels sont les avantages et inconvénients de XP ?*

Exercice 5 (Un peu de test en vrai)**.**

1- écrire une méthode Java (ou OCaml, ou autre) de signature

```
public static Vector unionSet (Vector a, Vector b)
```

retournant un vecteur d'objets contenus dans `a` *ou* `b`

2- Cet énoncé présente des défauts et des ambiguités. Trouvez-en autant que possible.

3- Créer un jeu de tests (CT/DT/oracle/script de test) pour détecter les bugs potentiellement introduits à cause de ces ambiguités / défauts de spécification. Documenter ces tests en décrivant de manière concise chacun d'entre eux. Jouer les tests contre votre implantation.

4- Réécrire la spécification pour lever les problèmes.