## R6.06 MAINTENANCE APPLICATIVE TD N°2 – Techniques Avancées par les Tests

### Objectifs de la feuille

- Analyse partitionnelle : technique de test fonctionnel
- Tests aux limites : technique de test fonctionnel
- Test syntaxique: technique de test fonctionnel
- Couverture du graphe de contrôle : test structurel dynamique
- Couverture sur flot de contrôle et flot de données

#### **Exercice 1** Analyse partitionnelle

On se propose de tester des programmes correspondants aux spécifications suivantes :

- 1. « Si la valeur n est négative : un message d'erreur est affiché. Si n est dans [1, 20 [, on affiche la valeur exacte de Factoriel(n). Si n est dans [20, 200] on affiche une approximation de Factoriel(n) en virgule flottante avec une précision de 0,1%. Si n > 200 un message d'erreur est affiché. »
- 2. « Écrire un programme qui calcule  $F(x)=(1/x)^{1/2}$  »
- 1.1 Donnez un jeu de test DT associé à chaque spécification.  $DT=\{-1;0;1;12;20;23;200;269\}$   $DT=\{-2;-1.5;0;1.5;12\}$

#### **Exercice 2** Test aux limites

Un programme de classification de triangles prend en entrée un triplet de réels (a,b,c) correspondants aux longueurs des 3 côtés d'un triangle. Le programme doit préciser la nature du triangle (équilatéral, isocèle, scalène, impossible).

- 2.1 Proposez un partitionnement des données en entrée pour tester cette procédure.
- 2.2 En déduire un jeu d'essai pour cette procédure.
- 2.3 Donnez des exemples de valeurs aux limites.
- 2.4 Ajoutez des exemples de valeurs aux limites aux jeux de test des spécifications de l'exercice précédent.

 $DTeq = \{a = 1; b = 1; c = 1\} \\ DTis = \{a = 2; b = 2; c = 3\} \\ DTsc = \{a = 5; b = 6; c = 7\} \\ DTim = \{a = 1; b = 1; c = 8\}$ 

DT = DTeq U DTis U DTsc U DTim





#### R6.06 MAINTENANCE APPLICATIVE TD N°2 – Techniques Avancées par les Tests

#### **Exercice 3** Test syntaxique

Soit l'interpréteur de commandes qui reconnaît des commandes du type :

- copy < fic1 > to < fic2 >
- rename  $\langle \text{fic1} \rangle$  to  $\langle \text{fic2} \rangle$

Où <fic1> et <fic2> sont des noms de fichiers formés sur une ou deux lettres prises dans l'ensemble {a,b}.

- **3.1** Donnez une grammaire
- 3.2 Donnez un arbre de dérivation 'générique'
- 3.3 Produisez des DT

#### **Exercice 4** Graphe Causes Effets

Un programme P prend en entrée une longueur (entier entre 1 et 20), une chaîne de caractères de cette longueur, et un caractère. P retourne sa position dans la chaîne ou un message d'erreur. Il est possible de cherche d'autres caractères. de cherche d'autres caractères.

chaîne de caractères - ET- position du caractère caractère présent —

- 4.1 Précisez les causes et les effets
- 4.2 Donnez le graphe Cause-Effet chaîne de caractères ET\_erreur 1<=longueur<=20 ~
- ET~chaîne de caractères 4.3 En déduire des jeux de test DT pas de caractère caractère présent

Exercice 5 Graphe de contrôle et chemins DT2={l=1;c="a";car="a"}

Soit le programme suivant :

```
lire(b,c,x);
     if (b < c) then
     begin
         d := 2*b;
         f := 3*c;
            (x >= 0) then
              y := x ;
              e := c ;
              if (y = 0) then
11
              begin
13
                  a := f-e ;
                  if (d < a) then
                  begin
                      writeln (a)
                  end
                  else
                      writeln (d)
21
                  end
              end
         end
23
```

DT1={l=5:c="hello":car="o"} DT3={l=20;c="constitutionnellement";car="u"} DT4={l=24;c="anticonstitutionnellement";car="u"}

- 5.1 Donnez le graphe de contrôle G
- 5.2 Donnez 3 chemins de contrôle du graphe G abcdefghik abk abcdefghik abcdk abcdeik
- 5.3 Donnez l'expression des chemins de contrôle de G
- 5.4 Soit DT1= $\{b=1,c=2,x=2\}$ . Donnez le chemin sensibilisé par DT1. abcdefk
- 5.5 On s'intéresse aux instructions ligne 16 et 20. Donnez des DT qui vont couvrir ces instructions.  $DT=\{b=1;c=8;x=0\}$ impossible
- 5.6 Donnez un chemin de contrôle non exécutable de G

## R6.06 MAINTENANCE APPLICATIVE TD N°2 – Techniques Avancées par les Tests

#### Exercice 6 Graphe de contrôle et chemins non exécutables

Soit le programme suivant :

```
lire (choix)
if (choix=1) then
    x := x+1;
if (choix=2) then
    x := x-1;
writeln(x);
```

- 6.1 Donnez le graphe de contrôle G correspondant au programme P4.
- **6.2** Donnez l'expression des chemins de contrôle de G.
- 6.3 En déduire le nombre de chemins de contrôle.
- 6.4 Donne le chemins de contrôle non exécutable.

#### **Exercice 7** Graphe de contrôle et chemins.

Soit l'algorithme suivant de recherche de l'emplacement d'un élément e dans un tableau T.

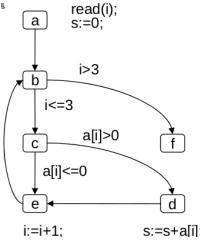
```
n := longueurT
i := 1
while i <= n et T[i] != e then
    begin
    do i := i + 1
    end;
if i <= n then
    begin
    return i
    end;
else
    begin
    return -1
    end;
end;
end;</pre>
```

- 7.1 Donnez le graphe de contrôle associé.
- 7.2 Donnez l'expression des chemins.
- 7.3 Dans le cas où le tableau a une taille de 3, donnez le nombre de chemins de contrôle.

```
abdeg abcbdeg abcbcbdeg abcbcbcbdeg abdfg abcbdfg abcbcbdfg abcbcbdfg
```

#### Exercice 8 Satisfaction d'un test structurel avec couverture

Soient le graphe de contrôle G suivant, deux chemins  $\delta 1 = \text{ cdebcde}$  et  $\delta 2 = \text{ ce}$ , et le test structurel  $T1 = \{\delta 1, \delta 2\}$ .



- 8.1 Le jeu de test DT = $\{a[1]=-2, a[2]=3, a[3]=-17, i=1\}$  satisfait-il T1?
- 8.2 Donnez l'expression des chemins de contrôle de G
- 8.3 En déduire le nombre de chemins de contrôle.

27



## R6.06 MAINTENANCE APPLICATIVE TD N°2 – Techniques Avancées par les Tests

#### **Exercice 9** Couverture TER1 et TER2 sur le flot de contrôle

```
lire (inf,sup);
i := inf;
sum := 0;
while (i <= sup) then
    begin
    do
        sum:=sum+a[i];
    i := i + 1
    end;
writeln(1/sum);</pre>
```

Soit le programme ci-contre qui calcule l'inverse de la somme des éléments d'indice entre inf et sup, d'un tableau tab contenant des entiers strictement positifs.

- 9.1 Testez le programme avec DT= $\{a[1]=1; a[2]=2; a[3]=3; inf=1; sup=3\}$ . Que se passe-t-il ?
- 9.2 Élaborez le graphe de contrôle et calculez TER1 et TER2 pour DT.

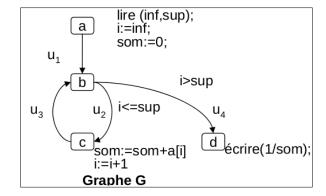
#### **Exercice 10** *Le nombre de Mac CABE*

```
if C1 then
    begin
    while (C2) then
        begin
        do X1;
        end;
    end;
else
    begin
    X2;
end;
X3;
```

- 10.1 Élaborez le graphe de contrôle.
- 10.2 Déterminez le nombre de nœuds dans le graphe
- 10.3 Déterminez le nombre d'arcs dans le graphe
- 10.4 En déduire le nombre de Mc CABE

#### Exercice 11 Le taux de couverture « tous-les-chemins-indépendants »

- 11.1 Donnez le nombre de chemins indépendants du graphe G.
- 11.2 Donnez une DT1 qui sensibilise M1=[abcbcbcbd].
- 11.3 Donnez une DT2 qui sensibilise M2=[abd].



11.4 Calculez le taux de couverture du critère tous-les-chemins-indépendants associé à DT1 U DT2.





## **Exercice 12** Séquence d'instructions entre deux branchements Soit le programme suivant :

```
005 INPUT A, C

010 B = 2*A

020 A = A+1

030 IF A < 0 THEN GOTO 60

040 B = -A

050 PRINT A+B

060 IF B = 2*C THEN GOTO 80

070 A = 1 : GOTO 90

080 A = -2 : GOTO 20

090 PRINT A

100 END
```

- 12.1 Donnez le graphe de contrôle.
- 12.2 Repérez les nœuds de type 'saut'.
- 12.3 Donnez les PLCS.

# **Exercice 13** Couverture TER3 du critère PLCS Soit le programme suivant :

```
main() {
    int i, factoriel;
    cin >> n;
    factoriel = 1;
    for (i = 1; i <= n; i++)
        factoriel = factoriel * i;
    printf("%d\n", factoriel);
}</pre>
```

- 13.1 Donnez le graphe de contrôle.
- 13.2 Donnez les PLCS.
- 13.3 Déterminez un DT afin que TER3=1

### **Exercice 14** Couverture sur flot de données

Soit le programme suivant :

```
lire(a, b)
if a < 0 then
begin
    b := b - a

    if b % 2 = 0 then
        begin
        p := b * b
        end
    else
        begin
        p := 2 * a
    end;
end;
writeln(p)</pre>
```

- 14.1 Quelles sont les variables définies?
- 14.2 Quelles sont les variables p-utilisées?
- 14.3 Quelles sont les variables c-utilisées?
- 14.4 Donnez un jeu de données pour un TER2 (tous les arcs)= 100 %
- 14.5 Couverture du critère toutes-les-définitions?