

MODULE --TEST ET VALIDATION

(chap4 Techniques de tests)

Exercices d'application

1. Techniques de test boîte noire : les partitions d'équivalence

Ex 1.1 :

Dans un examen un candidat doit avoir un minimum de 24 points pour qu'il soit déclaré admis . Le score max ne doit pas dépasser 40 points.

Q1 : Identifier les différentes partitions d'équivalence ainsi que les valeurs limites

Q2 : Quelle est la suite des valeurs de la partition d'équivalence valide qui permet à l'étudiant de réussir son examen :

- a) 22, 23, 26
- b) 21, 39, 40
- c) 29, 30, 31
- d) 0, 15, 22

Ex 1.2 :

Un programme valide les valeurs numériques selon la formule suivante :

- Les valeurs inférieures à 10 sont rejetées
- Les valeurs entre 10 et 21 sont acceptées
- Les valeurs supérieures ou égales à 22 sont rejetées

Q1 : déterminer les partitions d'équivalence

Q2 :Quelle est parmi les suites de valeurs suivantes celle qui couvre toutes les partitions équivalentes :

- a. 10, 11, 21
- b. 3, 20, 21
- c. 3, 10, 22
- d. 10, 21, 22

Q3 :Quelle est parmi les suites de valeurs suivantes celle qui couvre toutes les valeurs limites :

- a. 9,10,11,22
- b. 9,10,21,22
- c. 10,11,21,22
- d. 10,11,20,21

Ex.1.3

Dans un système conçu pour fonctionner sur l'impôt à payer:

- un employé qui touche un salaire de 4000 et moins ne paie pas d'impôt
- un employé qui touche un salaire pouvant atteindre 5500 doit payer les impôts à un taux de 10%
- un employé dont le salaire peut atteindre 33500 doit payer les impôts à un taux de 22%
- tout employé dont le salaire dépasse les 33500 doit payer les impôts à un taux de 40 %

Q1 : déterminer les partitions d'équivalence et les valeurs limites

Q2 : quelle est la suite des valeurs qui couvre 3 partitions d'équivalence :

- a) 4000; 5000; 5500
- b) 32001; 34000; 36500
- c) 28000; 28001; 32001
- d) 4000; 4200; 5600

2. Technique de test boîte blanche :

Couverture des instructions, couverture des conditions/décisions et couverture des branches

Ex.2.1 :

Soit l'exemple suivant :

```
Read A
IF (A = 8) THEN
  PRINT ("10")
ELSE
  PRINT (2*A)
```

1. Elaborer un graphe d'exécution du programme
2. Proposer des cas de test pour une couverture de 100 % des instructions

Ex.2.2 :

Soit le programme suivant

```
Déclaration des variables
Inf,sup,i ,som entier ;
Debut
Lire(inf,sup) ;
I :=inf ;
Som :=0
Tant que (i<=sup)
Faire
  Som :=som+i ;
  i :=i+1 ;
Fin tanque
Ecrire(1/som) ;
Fin
```

Travail demandé

1. Elaborer un graphe d'exécution du programme décrit en langage algorithmique
2. Concevoir des valeurs de tests qui permettent de faire le parcours complet des différents nœuds du graphe d'exécution avec 3 itérations dans la boucle

Ex.2.3

Soit le code suivant :

```
Lire a
Lire b
Si (a <0) ou (b<0) alors
  /**bloc négatif ***/
Sinon
  Si (((a<20) ou (a >=100)) et (b<=12)) alors
    /**bloc instruction BL1 ***/
  Sinon
    /**bloc instruction BL2 ***/
  Fin si
fin si
```

Travail demandé :

1. Construire le graphe d'exécution en nœud et branche
2. Déterminer les partitions d'équivalence et les valeurs limites
3. Construire une table de décisions
4. Proposer les cas de test pour assurer une bonne couverture de test des conditions /décisions

Ex.2.4

Dans les différents exemples suivants :

- Construire le graphe d'exécution
- Déterminer Le nombre minimum de cas de test pour les instructions
- Déterminer Le nombre minimum de cas de test pour les branches

1)

```
Read A
Read B
IF A > 0 THEN
  IF B = 0 THEN
    Print "No values"
  ELSE
    Print B
    IF A > 21 THEN
      Print A
    ENDIF
  ENDIF
ENDIF
ENDIF
```

2)

```
Read A
Read B
IF A < 0 THEN
  Print "A negative"
ELSE
  Print "A positive"
ENDIF
IF B < 0 THEN
  Print "B negative"
ELSE
  Print "B positive"
ENDIF
```

3)

```
Read A
Read B
IF A < 0 THEN
  Print "A negative"
ENDIF
IF B < 0 THEN
  Print "B negative"
ENDIF
```

4)

```
Read A
IF A < 0 THEN
  Print "A negative"
ENDIF
IF A > 0 THEN
  Print "A positive"
ENDIF
```