

Corrigé type de l'examen

Exercice 1:

1) Généralisation des cas de test selon la méthode de la table de décision :

a) les conditions:

C1: Un achat moins de 30 €.

C2: Un Achat plus ou égal 30 € et moins de 200 €

C3: Un Achat plus ou égal 200 € et moins 500 €

C4: Un Achat plus ou égal 500 € et moins 1000 €

C5: Un Achat plus ou égal 1000 €

C6: Une période de promotion

0,5

b) Les actions:

A1: Nouveau solde de points = ancien solde + 0

A2: Nouveau solde de points = ancien solde + Montant d'achat div 30

A3: Nouveau solde de points = ancien solde + Montant d'achat div 30 + 5

A4: Nouveau solde de points = ancien solde + Montant d'achat div 30 + 11

A5: Nouveau solde de points = ancien solde + Montant d'achat div 30 + 25

A6: Nouveau solde de points = ancien solde + Montant d'achat div 30 * 2

A7: Nouveau solde de points = ancien solde + (Montant d'achat div 30 + 5)*2

A8: Nouveau solde de points = ancien solde + (Montant d'achat div 30 + 11)*2

A9: Nouveau solde de points = ancien solde + (Montant d'achat div 30 + 25)*2

0,5

c) Les relations entre conditions et actions :

non(C6) et C1 => A1

non(C6) et C2 => A2

non(C6) et C3 => A3

non(C6) et C4 => A4

non(C6) et C5 => A5

C6 et C2 => A6

C6 et C3 => A7

C6 et C4 => A8

C6 et C5 => A9

1

c) Table de décision :

C1	1	0	0	0	0	0	0	0	0
C2	0	1	0	0	0	1	0	0	0
C3	0	0	1	0	0	0	1	0	0
C4	0	0	0	1	0	0	0	1	0
C5	0	0	0	0	1	0	0	0	1
C6	0	0	0	0	0	1	1	1	1
A1	1	0	0	0	0	0	0	0	0
A2	0	1	0	0	0	0	0	0	0
A3	0	0	1	0	0	0	0	0	0
A4	0	0	0	1	0	0	0	0	0
A5	0	0	0	0	1	0	0	0	0
A6	0	0	0	0	0	1	0	0	0
A7	0	0	0	0	0	0	1	0	0
A8	0	0	0	0	0	0	0	1	0
A9	0	0	0	0	0	0	0	0	1

d) Les cas de test

Cas de test id	Combinaisons (Période de promotion)	Données de test	Résultat attendus
1	Non	20 €	Nouveau solde de points = ancien solde + 0
2	Non	65 €	Nouveau solde de points = ancien solde + 6
3	Non	250 €	Nouveau solde de points = ancien solde + 24 + 5
4	Non	600 €	Nouveau solde de points = ancien solde + 60 + 11
5	Non	1200 €	Nouveau solde de points = ancien solde + 120 + 25
6	Oui	65 €	Nouveau solde de points = ancien solde + 6 * 2
7	Oui	250 €	Nouveau solde de points = ancien solde + (24 + 5)*2
8	Oui	600 €	Nouveau solde de points = ancien solde + (60 + 11)*2
9	Oui	1200 €	Nouveau solde de points = ancien solde + (120 + 25)*2

2) Généralisation des cas de test pour le cas d'utilisation « Attribuer une récompense de fidélité » en se basant sur la méthode de test aux limites :

ID de cas de test	Données de test	Résultat attendus
1	- Nombre de points = -1	Erreur
2	- Nombre de points = 0	- Pas de réduction - Pas de mise à jour des points
3	- Nombre de points = 1	- Pas de réduction - Pas de mise à jour des points
4	- Nombre de points = 20	- Pas de réduction - Pas de mise à jour des points
5	- Nombre de points = 49	- Pas de réduction - Pas de mise à jour des points
6	- Nombre de points = 50	- Réduction de 5% - Le solde de points est réinitialisé à 0
7	- Nombre de points = 51	- Réduction de 5% - Le solde de points est réinitialisé à 0
8	- Nombre de points = 60	- Réduction de 5% - Le solde de points est réinitialisé à 0
9	- Nombre de points = 69	- Réduction de 5% - Le solde de points est réinitialisé à 0
10	- Nombre de points = 70	- Réduction de 10% - Le solde de points est réinitialisé à 0
11	- Nombre de points = 71	- Réduction de 10% - Le solde de points est réinitialisé à 0
12	- Nombre de points = 90	- Réduction de 10% - Le solde de points est réinitialisé à 0
13	- Nombre de points = 119	- Réduction de 10% - Le solde de points est réinitialisé à 0
14	- Nombre de points = 120	- Réduction de 15% - Le solde de points est réinitialisé à 0
15	- Nombre de points = 121	- Réduction de 15% - Le solde de points est réinitialisé à 0
16	- Nombre de points = 170	- Réduction de 15% - Le solde de points est réinitialisé à 0
17	- Nombre de points = 199	- Réduction de 15% - Le solde de points est réinitialisé à 0
18	- Nombre de points = 200	- Réduction de 20%

4

		- Le solde de points est réinitialisé à 0
19	- Nombre de points = 201	- Réduction de 20% - Le solde de points est réinitialisé à 0
20	- Nombre de points = 300	- Réduction de 20% - Le solde de points est réinitialisé à 0

3) Le nombre minimal de tests pour garantir la couverture de toutes les paires de valeurs possibles en utilisant la méthode **Pairwise**:

Système d'exploitation	Navigateurs web	Modes de paiement
Windows	Firefox	espèces
Macintosh	Firefox	carte bancaire
Linux	Firefox	paiement mobile
Windows	Chrome	carte bancaire
Macintosh	Chrome	paiement mobile
Linux	Chrome	espèces
Windows	Edge	paiement mobile
Macintosh	Edge	espèces
Linux	Edge	carte bancaire

4) Le choix de Selenium avec Junit 5 est pertinent à condition que les deux outils soient utilisés dans leurs domaines respectifs.

Justification :

- **Selenium** est un outil d'automatisation des **tests fonctionnels** des **applications web**.

Dans le système de fidélité, il est utilisé pour :

- Tester l'interface utilisateur (UI) du site web,
- Simuler les actions d'un caissier (clics, saisie de montant, validation d'achat),
- Vérifier les résultats visibles (affichage des points, réduction appliquée, solde mis à jour).

Donc Selenium est adapté pour les **tests end-to-end (fonctionnels)**.

- **JUnit 5** est un framework de **test unitaire** pour **Java**.

Dans ce contexte, il sert à :

- Tester la logique métier du système de fidélité,
- Vérifier le calcul des points,
- Valider les règles de réduction (seuils, bonus),
- Exécuter rapidement des tests de non-régression sur les fonctions de calcul.

- Donc JUnit 5 est adapté pour les **tests unitaires** et **tests de logique métier**.

5) La différence entre le **re-test** et le **test de régression** :

a) **Re-test** :

Le **re-test** consiste à vérifier qu'un **défaut corrigé** a bien été résolu.

Objectif :

- Tester **uniquement le bug corrigé**

- Confirmer que la correction fonctionne correctement

Exemple dans le système de fidélité :

- Un bug empêchait l'ajout de **+11 points pour montant ≥ 500 €**
- Après correction, on reteste ce cas précis pour vérifier que les 11 points sont bien ajoutés

b) Test de régression :

Le **test de régression** consiste à vérifier que la correction d'un bug **n'a pas introduit de nouveaux problèmes ailleurs dans le système.**

Objectif :

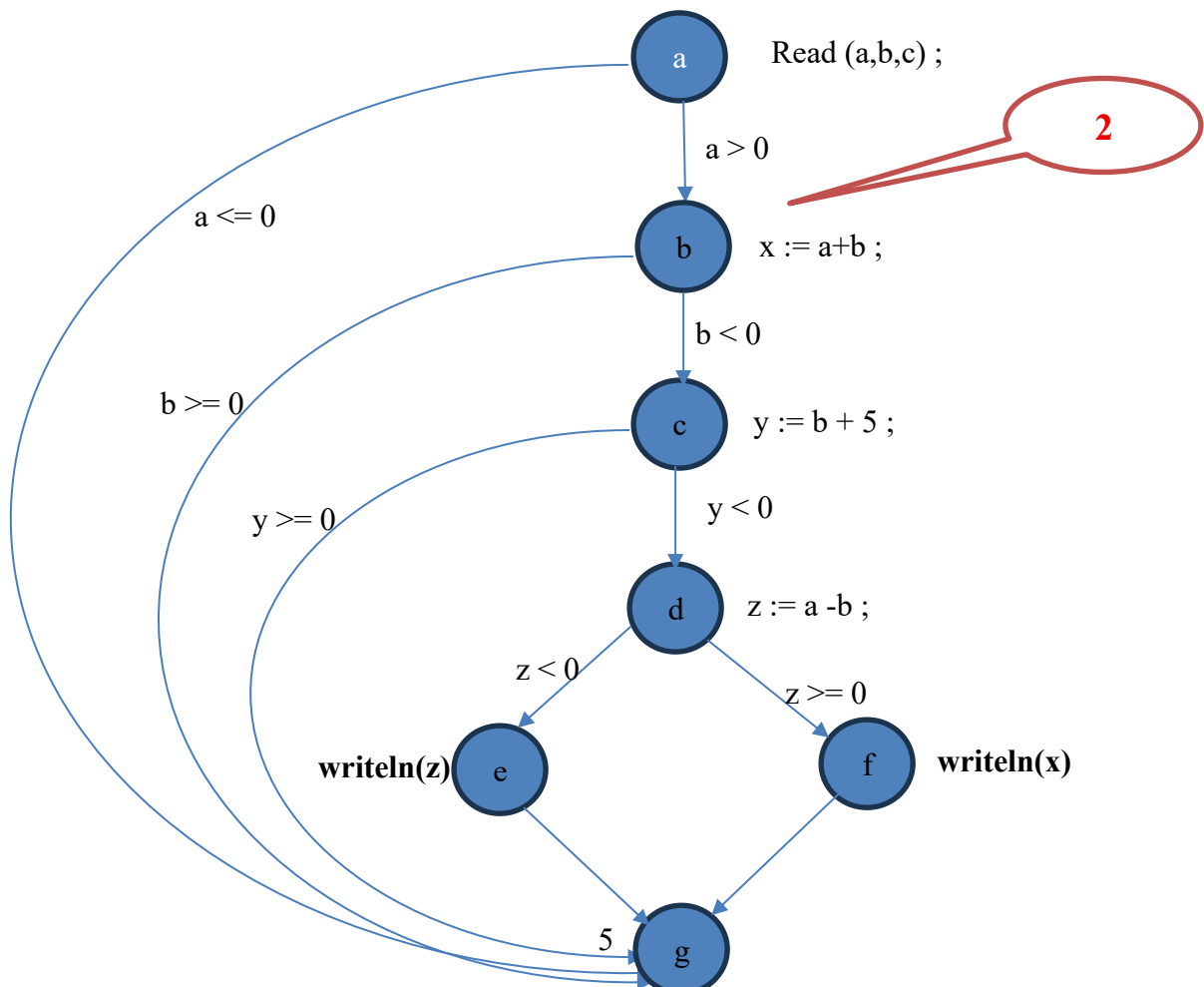
- Tester les fonctionnalités déjà existantes
- S'assurer que rien n'a été cassé après la modification

Exemple dans le système de fidélité :

- Après correction du bonus +11 :
 - Vérifier que le calcul des points normaux fonctionne toujours
 - Vérifier que les réductions (5%, 10%, 15%, 20%) sont toujours correctes

Exercice 2 :

1) Le graphe de contrôle G(P) associé au programme P :



2) L'expression des chemins de contrôle de G(P) :

$ag + abg + abcg + abcdeg + abcdg$

$= a(1+b+bc+bcd+bcdf)g$

$= a(1+b(1+c+cde+cdf))g$

$= a(1+b(1+c(1+de+df)))g$

$= a(1+b(1+c(1+d(e+f))))g$

1

3) Les chemins de contrôle non exécutable de G(P) :

On a : $z = a - b$. Pour exécuter cette instruction, on suppose que $a > 0$ et $b < 0$. Dans ce cas, z est toujours strictement positif. Par conséquent, le chemin **abcdeg** est un chemin non exécutable.

1

Exercice 3 :

1) Les données de test (DT) en fonction du critère de couverture de tous les arcs :

$DT1 = \{a=0, b=2\}$

$DT2 = \{a=5, b=2\}$

1

2) Les données de test (DT) en fonction du critère de couverture de toutes les

définitions :

$DT1 = \{a=0, b=2\}$

$DT2 = \{a=5, b=2\}$

1

3) Les données de test (DT) en fonction du critère de couverture de tous les utilisateurs :

$DT1 = \{a=0, b=2\}$

$DT2 = \{a=5, b=2\}$

$DT3 = \{a=-3, b=2\}$

$DT4 = \{a=4, b=2\}$

1