

**Question (3 points) 1 :** Le programme suivant doit être testé :

```

1. float Bisection(float a, float b, float eps){
2.     float xR=a;
3.     float xL=b;
4.     float xM=0.0;
5.     if (f(xL) * f(xR)>0)
6.         return ERROR;

7.     while ( abs(xR - xL) > eps){
8.         xM = (xR + xL) / 2
9.         if ((f(xL) * f(xM)) > 0)
10.            xL = xM
11.        else
12.            xR = xM
13.    }
14.    return xM;
15.}

16. float f(float x){ float tmp = (x*x-1); return tmp;}}
```

**Question 1.1 (1 point) :** Déterminez les last-defs et les first-uses pour tester l'interaction entre Bisection et f.

Variable	Last def	First use

### SOLUTION


Variable	Last def	First use
xR	2	13
xL	3,10	13
xM	8	13
tmp	13	9



**Question 1.2 (2 points) :** Pour la fonction Bisection donnez les valeurs du a, b et eps pour couvrir le critère all-coupling-defs call-coupling. Pour chaque valeur précisez la/les définition-use couverte(s) pour le critère all-coupling-defs call-coupling. Les données doivent être spécifiées dans le tableau suivant:

Cas de test	a	b	eps	DU couverte xR	xL	xM	tmp
T1							
T2							
T3							

### Solution

Cas de test	a	b	eps	DU couverte xR	xL	xM	tmp
T1	0.0	1.0	0.05	<2,13>	<3,13> <10,13> 	<8,13>	<13,9>

### Question 2 (5 points): Pour le programme suivant :

```

public class Item {
    private String name;
    private int price;

    public Item (String n, int p){
        name = n;
        price = p;
    }
    public void showData(){System.out.println("Item: "+name+ " Price " + price);}
    public String getPrice() {return price; }
    public void setPrice(int p) {price = p; }
}

public class DiscountedItem extends Item {
    private int discount;

    public DiscountedItem (String n, int p, int d){
        super(n, p);
        discount = d;
    }

    public void setDiscount(int d) {discount = d;}
    public int applyDiscount( ) { price = price - discount*price/100; return price; }
}

```

**Question 2.1 (1 point) :** Si on utilise le critère de Harrold McGregor, quelles sont les méthodes du DiscountedItem qui doivent être testées de nouveau? Donnez les résultats en remplissant le tableau suivant :

Méthode	A retester (oui ou non) ?
setDiscount	
applyDiscount	
showData	
getPrice	
setPrice	

### SOLUTION

Méthode	A retester (oui ou non) ?
setDiscount	oui
applyDiscount	oui
showData	oui
getPrice	oui
setPrice	oui

**Question 2.2 (1 point) :** Pour les classes Item et DiscountedItem, complétez le tableau suivant pour le MaDUM, ajouter des ajouter des reporters (et leurs setters)” si nécessaire (t transformer, c constructor, o other, et r reporter) ; voir l’exemple pour Item(n,s) qui est « constructor »; les données doivent être spécifiées dans le tableau suivant:

	Item (..)	DiscountedItem (..)								
name	c	c								
price	c	c								
discount	c	c								

### SOLUTION



	Item (..)	DiscountedItem (..)	set Price	get Price	set Name	get Name	show Data	get Discount	set Discount	apply Discount
name	c	c			t	r	o			
price	c	c	t	r			o			t
discount	c	c						r	t	o

**Question 2.3 (1 point) :** Identifiez les tranches pour les attributs `name`, `price`, et `discount`; les données doivent être spécifiées dans le tableau suivant:

Attribute	Tranche							
name								
price								
discount								

### SOLUTION

Attribute	Tranche							
name	setName	getName	showData					
price	setPrice	getPrice	showData	applyDiscount				
discount	setDiscount	getDiscount	applyDiscount					

**Question 2.4 (2 points) :** Pour `DiscountedItem` et la tranche `Tranche(price)`, donnez les séquences du test des transformers dans le tableau suivant:

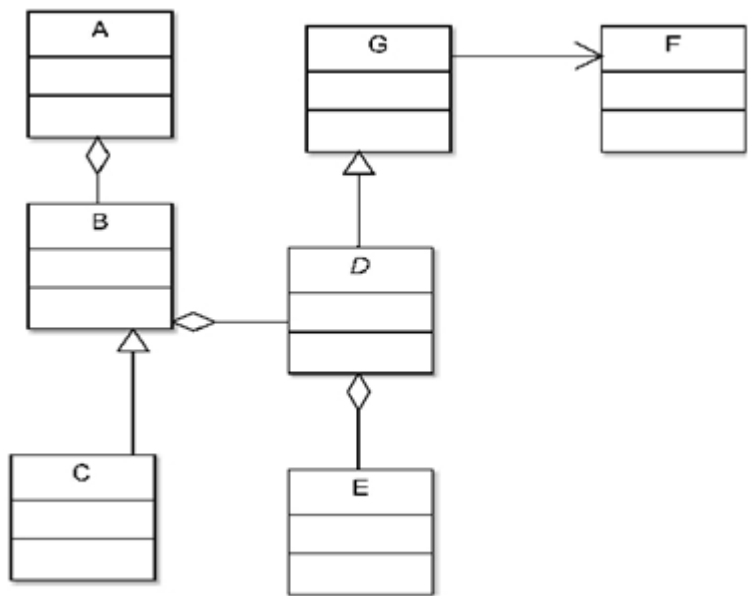
Séquences du test pour <code>Tranche(price)</code>								

### SOLUTION



Séquences du test pour Tranche(price)								
	DiscountedItem (..)	setPrice	getPrice					
	DiscountedItem (..)	applyDiscount	getPrice					
	DiscountedItem (..)	setPrice	applyDiscount	getPrice				
	DiscountedItem (..)	applyDiscount	setPrice	getPrice				

**Question 3 (3 points) :** Pour le diagramme UML suivant :



**Question 3.1 (1 point)** Calculez le coupe-feu; donnez les résultats en remplissant le tableau suivant :

Classe X	coupe-feu(X)
A	
B	
C	
D	
E	
F	
G	

**SOLUTION**

Classe X	coupe-feu(X)
A	-
B	A, C
C	-
D	A, B, C
E	A, B, C, D
F	A, B, C, D, G
G	A, B, C, D

**Question 3.2 (2 points) :** Calculez l'ordre optimal d'intégration; donnez les résultats en remplissant le tableau suivant :

Niveau de test	Classes
1	
2	
3	
4	
5	
6	

### SOLUTION

Niveau de test	Classes
1	E, F
2	G
3	D
4	B
5	A, C

**Question 4 (8 points) :** Le tableau suivant (TAB 1) contient les défaillances enregistrées (en sec.) pour un système réel :

211	2047	3924	5822	8978
366	2421	4113	6138	9111
608	2911	4230	7123	9243
677	3035	4776	7768	9578

998	3089	5003	8061	10523
1278	3565	5126	8191	11806
1288	3623	5467	8323	12185
1535	3803	5690	8723	14729

Pour le model Musa de base, la relation entre les défaillances trouvées et l'intensité des défaillances est :

$$\lambda = \lambda_0 \left( 1 - \frac{\mu}{v_0} \right)$$

Pour k=4 déterminez :

**Question 4.1 (4 points) :** les couples (m,r) et les valeurs de  $\lambda_0, v_0$  pour les données de TAB 1; les données doivent être spécifiées dans le tableau suivant:

Init	End	Delta	r	m

### SOLUTION

211	2047	3924	5822	8978
366	2421	4113	6138	9111
608	2911	4230	7123	9243
677	3035	4776	7768	9578
998	3089	5003	8061	10523
1278	3565	5126	8191	11806
1288	3623	5467	8323	12185
1535	3803	5690	8723	14729

Tableau du temps, défets, etc :

Init	End	Delta	r	m
0	677	677	0.005908	0
677	1535	858	0.004662	4
1535	3035	1500	0.002667	8
3035	3803	768	0.005208	12
3803	4776	973	0.004111	16
4776	5690	914	0.004376	20
5690	7768	2078	0.001925	24

7768	8723	955	0.004188	28
8723	9578	855	0.004678	32
9578	14729	5151	0.000777	36

Si  $y=a+bx$  et  $x=m$ ,  $y=r$ ,  $b$  est la solution d'une régression linéaire

$$b=-7.28002E-05 \quad a= 0.005160514$$

$$\lambda_0=0.0052 \quad \nu_0=70.88=71$$

**Question 4.2 (1 point) :** Combien de défauts sont encore présents dans le logiciel ?

SOLUTION

On a trouvé 40 défauts et on estime qu'il en reste encore 31 dans le logiciel.

**Question 4.3 (1 point)** Si le logiciel est de 30000 LOC, quels sont les pourcentages de défauts initial et final?

SOLUTION

On a trouvé 40 défauts et on estime qu'il reste encore 31 défauts dans le logiciel. Les pourcentages de défauts initial et final sont  $71/30000$  et  $31/30000$  i.e. 2.4 et 1 pour KLOC.

**Question 4.4 (2 points) :** Si la valeur de  $\lambda$  requise est d'une (1) défaillance pour une journée (on n'accepte pas plus qu'une défaillance par 24 heures), déterminez les défaillances à corriger et le temps CPU des activités de test requis pour s'assurer de respecter la valeur de défaillance requise. Si pour une heure de temps CPU, sont requises 20 heures de travail du personnel, combien de personnes-heures sont requises pour obtenir la valeur voulue?

SOLUTION

Avec  $\lambda_0 = 0.0052, \nu_0 \cong 71$  on a un taux de défaillances observé de 0.000776548 CPU/sec, c'est-à-dire  $60*60*24*0.000776548 = 67$  défaillances pour 24 heures **or** on vise un taux de défaillances de  $1/86400=1.15741E-05$  défaillances pour sec. Les défaillances à corriger sont donc :

$$\Delta\mu = \frac{\nu_0}{\lambda_0} (\lambda_p - \lambda_f) = 10.05 \cong 10$$

le temps requis est



$$\Delta t = \frac{\nu_0}{\lambda_0} \lg(\lambda_p / \lambda_f) = 25091.73251 \cong 7$$

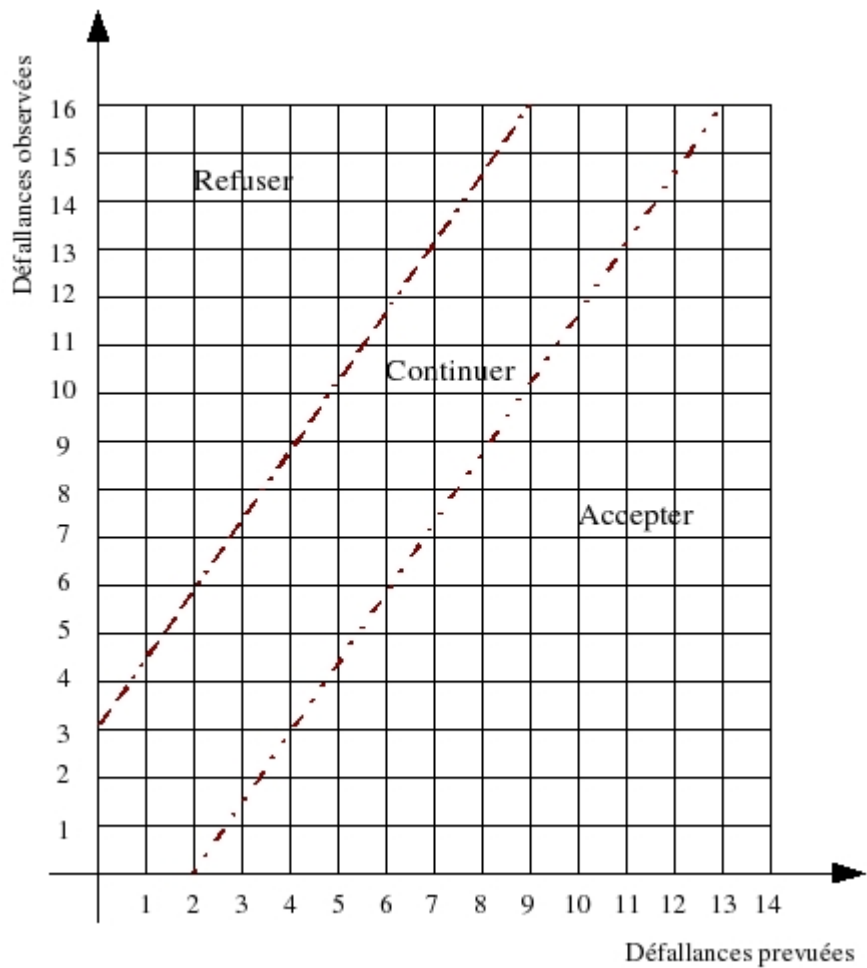
25091 sec = 7 heures de temps CPU. Pour chaque heure CPU, 20 heures de travail sont requises donc 7\*20= 140 heures ce qui équivaut à entre 1 mois de travail (40 heures per semaine per personne).

**Question 5 (1 point) :** On a effectué des activités de test sur un système téléphonique logiciel de 250 KLOC ; on a observé les défaillances :

Défaillance	Million d'opérations
1	0.20
2	0.80
3	2.5
4	3.1
5	6.5

Si on requiert au plus une (1) défaillance pour 1 million d'appels, utilisez la carte de décision suivante pour décider si on « accepte » le logiciel, si on le « refuse » ou si on continue les activités de test :

Défaillance	Million d'opérations	Défaillances Prévues	décision
1	0.20		
2	0.80		
3	2.5		
4	3.1		
5	6.5		



### SOLUTION

Défaillance	Million d'opérations	Def Prévues	décision
1	0.20	0.20	Cont
2	0.80	0.80	Cont
3	2.5	2.5	Cont
4	3.1	3.1	Cont
5	6.5	6.5	Accepter