

Exercice 1 (7 points) : Ce programme calcule la valeur en dollars et en cents d'un ensemble de pièces de différentes dominations entrées par l'utilisateur. Les pièces sont : 1, 5, 10, 25, 50, 100. Malheureusement, il contient plusieurs erreurs. Trouvez-les et corrigez-les !

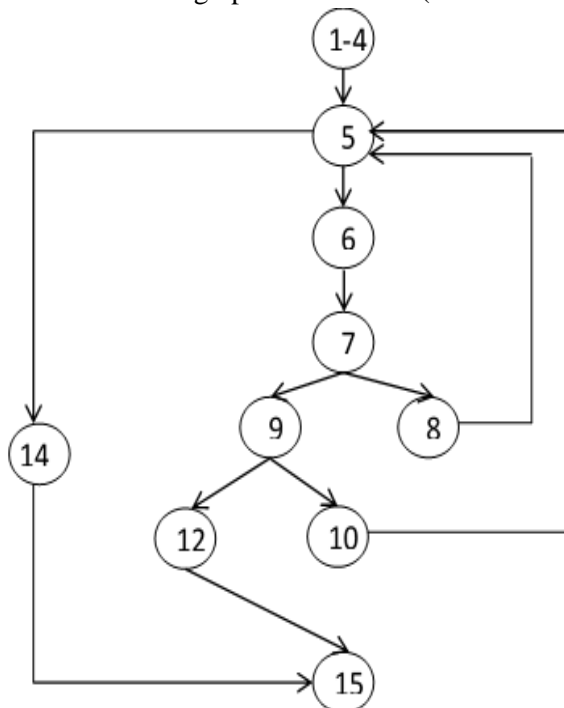
```

main ()
{
int total_coin_value;           int total_coin_value=0;           1 pt
int number_of_coins = 0;
int number_of_dollars = 0;
int number_of_cents = 0;
int coin_values = {1,5,10,25,25,100};    {1,5,10,25,50,100};           0,5 pt
{                                   Accolade à supprimer           1 pt
int i = 1;                           int i = 0                       1 pt
while ( i < 6)
{
printf("input number of coins\n");
scanf ("%d", number_of_coins);    scanf ("%d", &number_of_coins);  1 pt

total_coin_value = total_coin_value +
(number_of_coins * coin_value[i]);    (number_of_coins * coin_value[i]);  0,5 pt
}
i = i + 1;                           i = i + 1; à l'intérieur de la boucle 1 pt
number_of_dollars = total_coin_value/100;
number_of_cents= total_coin_value - (100 * number_of_dollars); number_of_cents 1pt
printf("%d\n", number_of_dollars);
printf("%d\n", number_of_cents);
}
    
```

Exercice 2 (7 points) : Soit le programme suivant :

1. Dressez son graphe de contrôle (utilisez les numéros des lignes) 2 pt



```

1. int binsearch (int x,int v[], int n){
2. int low, high, mid;
3. low = 0;
4. high = n-1;
5. while (low <= high) {
6. mid = (low+high)/2
7. if (x < v[mid])
8. high = mid-1;
9. else if (x > v[mid])
10. low = mid + 1;
11. else
12. return mid;
13. }
14. return -1;
15. }
    
```

2. Calculez sa complexité cyclomatique 0,5 pt

$$\begin{aligned}
 \text{Complexité cyclomatique} &= \text{Nombre de conditions} + 1 = 3 + 1 = 4 \\
 &= \text{Nbre arcs} - \text{Nbre Nœuds} + 2 = 12 - 10 + 2 = 4
 \end{aligned}$$

3. Combien de cas de test sont nécessaires pour couvrir adéquatement le code par rapport à ses branches ?

3 cas de test sont nécessaires :

1-4, 5, 6, 7, 9, 10, 5, 14, 15

1-4, 5, 6, 7, 8, 5, 14, 15

1-4, 5, 6, 7, 9, 12, 15

0,5 pts

4. Dressez le tableau des définitions et des utilisations (c-uses et p-uses) des variables manipulées dans ce programme ainsi que le tableau tous les du-paths.

	Définition	c-use	p-use
1	x, v, n		
2			
3	low		
4	high	n	
5			low, high
6	mid	low, high	
7			x, v, mid
8	high	mid	
9			x, v, mid
10	low	mid	
11			
12		mid	
13			
14			
15			

1,5 pts

Ligne de debut - > ligne de fin	c-use	p-use
1-7		x
1-9		x
1-7		v
1-9		v
1-4	n	
3-5		low
4-5		high
3-6	low	
4-6	high	
6-7		mid
6-8	mid	
6-9		mid
6-10	mid	
6-12	mid	

1,5 pts

5. Par rapport à la variable (mid), quel est nombre de cas de test nécessaires pour couvrir toutes ses définitions, tous ses c-uses et tous ses p-uses.

	Nbre de cas de test pour mid
Toute les définitions	1
Tous le c-uses	3
Tous les p-uses	2
Tous les du-path	5

1 pt

Exercice 3 (7 points) : Une application lit deux caractères : le premier est un caractère alphabétique qui doit être "A" ou "B" et le deuxième est un chiffre. En fonction de ces deux caractères, des messages doivent être affichés.

- Si le premier caractère lu est un "A" ou un "B" et le deuxième caractère est un chiffre, le message « *le fichier doit être mis à jour* » est affiché.
- Si le premier caractère est incorrect (n'est ni "A" ni "B"), le message « *Caractère attendu* » est affiché.
- Si le deuxième caractère est incorrect (n'est pas un chiffre), le message « *Chiffre attendu* » est affiché.

1. Identifier les causes et les effets

Les causes sont :

- C1 - Le premier caractère est A
- C2 - Le premier caractère est B
- C3 - Le deuxième caractère est un chiffre

1 pt

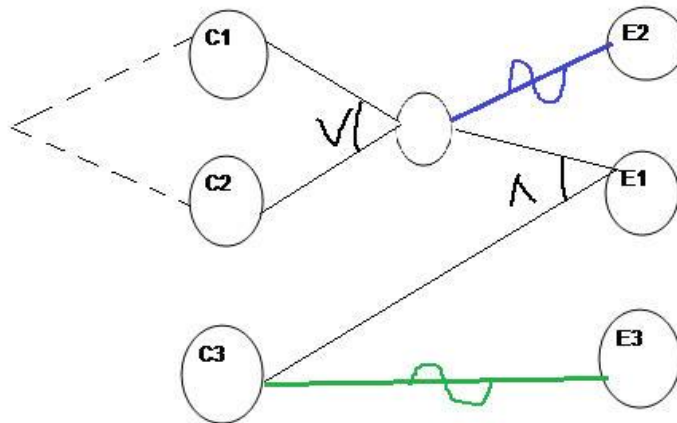
Les effets sont :

- E1 - le message « le fichier doit être mis à jour » est affiché.
- E2 - le message « Caractère attendu » est affiché.
- E3 - le message « Chiffre attendu » est affiché
- E4 – Situation impossible

1 pts

2. Tracer le graphe causes-effets

2 pts



3. Dresser la table de décision

2 pts

	1	2	3	4	5	6	7	8
C1	T	T	T	T	F	F	F	F
C2	T	T	F	F	T	T	F	F
C3	T	F	T	F	T	F	T	F
E1	F	F	T	F	T	F	F	F
E2	F	F	F	F	F	F	T	T
E3	F	F	F	T	F	T	F	F
E4	T	T	F	F	F	F	F	F

4. Peut-on minimiser le nombre de cas de test. Si oui, montrez comment

Les deux premières colonnes ainsi que les deux dernières peuvent être regroupées ensemble. On obtient la table suivante.

1 pt

	1	3	4	5	6	7
C1	T	T	T	F	F	F
C2	T	F	F	T	T	F
C3	-	T	F	T	F	-
E1	F	T	F	T	F	F
E2	F	F	F	F	F	T
E3	F	F	T	F	T	F
E4	T	F	F	F	F	F

Bon courage !