

# Flot de contrôle

Florian Bourse

Considérons le code suivant :

```
void triangle(int j, int k, int l) {
    assert(j > 0 && k > 0 && l > 0);
    int eg = 0;
    if (j + k <= l || k + l <= j || l + j <= k) {
        printf("impossible");
    }
    else {
        if (j == k) {
            eg = eg + 1;
        }
        if (l == k) {
            eg = eg + 1;
        }
        if (l == j) {
            eg = eg + 1;
        }
        if(0 == eg) {
            printf("scalene");
        } else {
            if (1 == eg) {
                printf("isocele");
            } else {
                printf("equilateral");
            }
        }
    }
}
```

1. Que fait cette fonction ?
2. Déterminer le graphe de flot de contrôle de cette fonction.
3. Donner le chemin dans le graphe correspondant à l'exécution de la fonction sur l'entrée :  $j = 3, k = 5, l = 3$ .
4. Construire un jeu de tests réalisant une couverture des branches.
5. Trouver un chemin du graphe non faisable.
6. Rajouter (si besoin) des tests pour que les conditions soient testées de manière exhaustive.
7. Compléter cette fonction pour tester si le triangle est rectangle.

```
int fastexp(int a, int b) {  
    int res = 1;  
    while (0 != b) {  
        if (1 == b % 2) {  
            res = a * res;  
        }  
        a = a * a;  
        b = b / 2;  
    }  
    return res;  
}
```

1. Déterminer le graphe de flot de contrôle de cette fonction.
2. Proposer un jeu de tests réalisant une couverture des branches.
3. Proposer un jeu de tests pour les valeurs aux limites.
4. Proposer une modification pour éviter les erreurs et les boucles infinies.