

# TD vérification logicielle

## Exercice 1

1. Soit la fonction suivante :

```
public int square(int n) { // n >= 0
    int cpt = n;
    int r = 0;
    while (cpt > 0) {
        r = r + n;
        cpt = cpt - 1;
    }
    return r;
}
```

- (a) Que calcule cette fonction ?  
(b) Écrivez le triplet de Hoare correspondant à cette fonction.  
(c) Déterminez l'invariant de boucle.  
(d) Démontrez la validité du triplet en appliquant les règles de déduction de la logique de Hoare.
2. Soit la fonction suivante :

```
public int isqrt(int n) { // n >= 0
    int r = 0;
    while (n >= (r+1)*(r+1)) {
        r = r+1;
    }
    return r;
}
```

- (a) Mêmes questions que précédemment.
3. Soit la fonction suivante :

```
public int fact(int n) { // n >= 0
    int t = n;
    int r = 1;
    while (t != 0) {
        r = r * t;
        t = t - 1;
    }
    return r;
}
```

(a) Mêmes questions que précédemment.

4. Soit la fonction suivante :

```
public int sum(int n, int [] t) { // n >= 0
    int j = 0;
    int s = 0;
    while (j < n) {
        s = s + t[j];
        j = j + 1;
    }
    return s;
}
```

(a) Mêmes questions que précédemment.

5. Soit la fonction suivante :

```
public int pow(int a, int n) { // a >= 0 et n >= 0
    int b = a;
    int m = n;
    int r = 1;
    while (m > 0) {
        if (m % 2 == 0) {
            b = b * b;
            m = m / 2; // division entiere
        } else {
            r = r * b;
            m = m - 1;
        }
    }
    return r;
}
```

(a) Mêmes questions que précédemment.

6. Soit la fonction suivante :

```
public int mult(int x, int y) { // x >= y >= 0
    int r = 0;
    int a = x;
    int b = y;
    while (b > 0) {
        if (b % 2 == 1) {
            r = r + a;
        }
        a = 2 * a;
        b = b / 2; // division entiere
    }
    return r;
}
```

(a) Mêmes questions que précédemment.