

# Variants

Florian Bourse

```
bool palindrome(char str[]){
    int i = 0,
        j = 0;

    while (str[j] != '\0') { // Loop 1
        j = j + 1;
    }

    j = j - 1;

    while (j >= i) { // Loop 2
        if (str[i] != str[j]) {
            return false;
        }
    }

    return true;
}
```

```
int log2(int x) {
    int l = 0;

    while (x > 0) {
        x = x / 2;
        l = l + 1;
    }

    return l;
}
```

```
int f(int x) {
    int n = 0;

    while (x > 0) {
        x = -2 * x + 10;
        n = n + 1;
    }

    return n;
}
```

```

int pgcd(int a,int b) {
    while (a != b) {
        if (a > b) {
            a = a - b;
        } else {
            b = b - a;
        }
    }

    return a;
}

```

```

int sqrt(int n) {
    int c = 0,
        s = 1;

    while (s <= n) {
        c = c + 1;
        s = s + 2 * c + 1;
    }

    return c;
}

```

```

int mystere(int p) {
    int c = 0;

    while (p > 0) {
        if (c == 0) {
            p = p - 2;
            c = 1;
        } else {
            p = p + 1;
            c = 0;
        }
    }

    return p;
}

```

Exercice : Écrire une fonction `void swap(int* tab, int i, int j)` qui échange les valeurs de `tab[i]` et `tab[j]`.

Écrire une fonction `void trie(int* tab, int n)` pour trier un tableau `tab` de taille `n` en utilisant l'algorithme suivant : pour chaque case en partant de 0, on y met le minimum des nombres restant à trier.