1 Problématique

Un serveur *World of Warcraft* représente 5,5 milions de lignes de code, *Windows 7* 40 millions et *Facebook* plus de 62 millions.

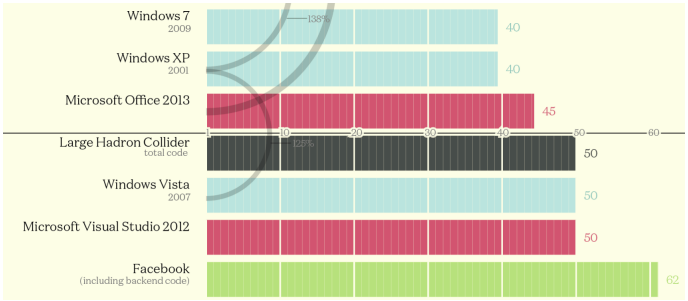

FIGURE 1 – Infographie complète sur informationisbeautiful

Il est impératif que chaque bloc de code (boucle, fonction...) réalise correctement la tâche qui lui est assignée.

Quels systèmes de contrôle peut-on mettre en place dans un programme informatique ?

2 Variant de boucle : terminaison

```
1 i = 10
2 while i >= 0:
3     print(i)
4 print("Boum")
```

Code 1 – Boucle infinie

Le code 1 ne termine jamais. Nous pouvons supposer que ce n'était pas la volonté initiale du programmeur.

À retenir

Un *variant de boucle* est une expression dont la valeur varie à chaque itération de la boucle.

Pour chaque boucle construite, il est donc nécessaire de déterminer son *variant* afin de prouver sa **terminaison**.

```
1 i = 10
2 while i >= 0:
3     print(i)
4     i = i - 1
5 print("Boum")
```

Code 2 – Compte à rebours

Christophe Viroulaud

Dans le code 2, i est un variant de la boucle : à chaque itération sa valeur décroît de 1, elle deviendra donc négative.

Activité 1 : L'algorithme 3 permet de déterminer si *mot* est un palindrome.

```
Fonction: palindrome(mot)
Debut
i = 0
j = longueur(mot) -1
TantQue i <= j faire
Si mot_i = mot_j alors
       i = i + 1
       j = j -1
sinon
       Renvoyer Faux
Fin Si
Fin TantQue
Renvoyer Vrai
Fin
```

Code 3 – Palindrome

1. Sur papier, dérouler l'algorithme pour les mots *radar* puis *sauras*.
2. Écrire cet algorithme en Python.
3. Montrer que l'expression $\delta = j - i$ est un variant de la boucle.

3 Invariant de boucle : correction

Une boucle peut terminer mais cependant ne pas réaliser ce à quoi elle est destinée.

À retenir

Un *invariant de boucle* est une propriété qui si elle est vraie avant l'itération, le reste après son exécution.

```
def somme_des_entiers(n):
    s = 0
    for i in range(1, n+1):
        s += i
    return s
```

Code 4 – Somme des premiers entiers

Le code 4 calcule la somme $1 + 2 + ... + n$. Nous pouvons montrer par récurrence que l'expression

$$s = 1 + 2 + ... + i - 1$$

Christophe Viroulaud

est un invariant de la boucle :

— Avant la première itération, $i = 1$ et $s = 0$; l'expression est vérifiée.

— On suppose que la condition est vérifiée au rang i.

— On montre alors qu'elle est vraie au rang $i + 1$.

Remarque

À la dernière itération, $i = n + 1$; la boucle n'est alors pas exécutée et l'invariant est bien vérifié :

$$s = 1 + 2 + ... + (n + 1) - 1$$
$$s = 1 + 2 + ... + n$$

Activité 2 : La division de l'entier a par l'entier b définie par Euclide consiste à trouver deux entiers q et r tels que

$$a = q \times b + r$$
$$\text{avec } 0 \leq r < b$$

Le code **??** détermine ces deux entiers.

```
1 def division(a, b):
2     q = 0
3     r = a
4     while r >= b:
5         q += 1
6         r _= b
7     return (q, r)
```

Code 5 – Division euclidienne

1. Dérouler à la main la fonction pour $a = 20$ et $b = 6$.
2. Montrer que l'expression $a = q \times b + r$ est un invariant de la boucle.

4 Préconditions

5 Tests

Christophe Viroulaud