

□ Exercice 1 : nombre de chiffres d'un entier positif

1. Écrire une fonction `int nb_chiffres(int n)` qui prend en paramètre un entier positif et renvoie le nombre de chiffres de cet entier. Par exemple, pour l'entier 1942, la fonction renvoie 4. Cette fonction *doit* procéder de la façon suivante :
 - si `n` vaut 0 alors on renvoie 1
 - sinon on divise `n` par 10 jusqu'à ce que `n` soit égal à 0, en incrémentant un compteur à chaque division. On renvoie la valeur de ce compteur lorsque le processus se termine.

```
1  int nb_chiffres(int n)
2  {
3      assert (n>=0);
4      if (n==0) {return 1;}
5      int c = 0;
6      while (n!=0)
7      {
8          c = c + 1;
9          n = n / 10;
10     }
11     return c;
12 }
```

2. Proposer un jeu de tests sous forme d'instructions `assert` permettant de valider le comportement de cette fonction.

```
1  assert(nb_chiffres(0) == 1);
2  assert(nb_chiffres(5) == 1);
3  assert(nb_chiffres(1942) == 4);
4  assert(nb_chiffres(99999) == 5);
5  assert(nb_chiffres(100000) == 6);
```

3. Prouver la terminaison de la fonction `nb_chiffres`.

Si `n` vaut 0 la fonction termine immédiatement en renvoyant 1. Sinon, montrons que `n` est un variant de la boucle `while (n > 0)` :

- `n` est un entier par précondition et strictement positif par condition d'entrée dans la boucle
- à chaque itération, `n` est divisé par 10 et comme il est strictement positif, il diminue strictement.

Donc cette fonction termine toujours.