

## 🔑 Mesurer le temps en C

Pour mesurer des temps en C, on utilise la librairie `time.h` (on prendra garde à ne pas oublier `##include<time.h>`). Celle-ci fournit notamment les fonctions suivantes :

- la fonction `time` donne le temps en secondes écoulé depuis un instant fixé (le 1er janvier 1970, mais peu importe). Elle prend en argument un pointeur dans lequel enregistrer le résultat. Ce résultat étant aussi retourné, on peut passer `NULL` en argument. Son type de retour est `time_t`, ce qui correspond au type `int` ou `long int`.
- la fonction `clock` donne le temps en **microsecondes** écoulé depuis un instant fixé (le début de l'exécution du programme). Elle ne prend aucun argument et son type de retour est `time_t`, ce qui correspond aussi au type `int` ou `long int`.

Dans les deux cas on obtient une par différence entre les instants de début et de fin.

- Si on utilise la fonction `time`, la précision est de l'ordre de la seconde. Si on souhaite convertir ce nombre entier de secondes en une valeur de type `double`, par exemple en vue de calculer un temps moyen par division (réelle!), on utilise la fonction `difftime` qui permet à la fois de faire la différence entre les deux instants de type `time_t` et de la convertir en `double`.
- Si on utilise la fonction `clock`, la précision est de l'ordre de la microseconde, on peut afficher le résultat comme un nombre décimal de secondes grâce une division réelle.

**Exemple.** On propose deux fonctions pour étudier `fibonacci` (volontairement mal codée). En regard d'un main appelant ces deux fonction, on présente l'affichage obtenu.

```
1 | int fibo_bete (int n){
2 |     if ( n == 0 || n == 1){return 1;}
3 |     else {return fibo_bete (n-1) + fibo_bete (n-2);}
4 | }

1 | void fibo_affiche_tps (int n){
2 |     time_t deb = time (NULL);
3 |     fibo_bete(n);
4 |     time_t fin = time (NULL);
5 |     double duree = difftime(fin,deb);
6 |     printf("n=%d : %fs\n",n,duree);
7 | }

1 | void fibo_affiche_tps_bis (int n){
2 |     clock_t deb = clock ();
3 |     fibo_bete(n);
4 |     clock_t fin = clock ();
5 |     clock_t duree = fin - deb;
6 |     float duree_s = duree/1000000.0;
7 |     printf("n=%d : %dms",n, duree);
8 |     printf(" soit %fs\n",duree_s);
9 | }
```

```
1 | printf("%d ----\n", clock());          2100 ----
2 | fibo_affiche_tps(36);                  n=36 : 1.000000s
3 | fibo_affiche_tps(38);                  n=38 : 2.000000s
4 | fibo_affiche_tps(40);                  n=40 : 6.000000s
5 | printf("-----\n");
6 | printf("%d \n", clock());              8770421
7 | printf("-----\n");
8 | fibo_affiche_tps_bis(36);              n=36 : 840657ms soit 0.840657s
9 | fibo_affiche_tps_bis(38);              n=38 : 2189553ms soit 2.189553s
10 | fibo_affiche_tps_bis(40);              n=40 : 5752209ms soit 5.752209s
11 | printf("-----\n");
12 | printf("%d \n", clock());              17553033
```