

## Temps de réaction

La réalisation de ce TP est conditionnée par la maîtrise des notions suivantes :

- structures de contrôle ;
- création et appel de fonction ;
- génération de valeurs aléatoires.

Votre objectif est de réaliser un programme qui va vous permettre de mesurer le temps de réaction d'un individu. Nous avons toujours entendu dire qu'il était en moyenne d'une seconde pour un adulte, nous allons vérifier cela dans le contexte d'un réflexe simple face à un stimulus. L'objectif de ce programme est de générer un signal visuel (stimulus) et de mesurer le temps que l'utilisateur a mis avant d'appuyer sur une touche du clavier (réaction). Voici le déroulement précis du programme.

- Lors du lancement du programme, vous devrez afficher un compte à rebours qui va permettre de préparer l'utilisateur à réagir au stimulus.
- À la fin de ce compte à rebours, un temps d'attente aléatoire devra être établi avant d'afficher le stimulus. Le côté aléatoire de ce temps empêche l'utilisateur de connaître précisément le moment de l'apparition du stimulus.
- Le stimulus est alors composé de caractères choisis apparaissant dans la console. Nous fixons un temps de réaction limite (par exemple 3 secondes) au delà duquel le programme s'arrête et affiche un message indiquant que l'utilisateur est soit endormi, soit vraiment lent. Durant ces  $N$  secondes, le programme affiche à l'écran au maximum 50 caractères, à raison de 1 caractères tous les  $N/50$  secondes. Cela permet de visualiser clairement le temps de réaction humain.
- Durant cet affichage, le programme se prépare à recevoir la réaction de l'utilisateur via la saisie d'une touche du clavier.
- Lorsqu'une touche est saisie (n'importe laquelle), le programme cesse son affichage et indique à l'utilisateur son temps de réaction.

Pour mener à bien cet exercice, vous allez donc créer les fonctions suivantes.

**L'usage de la fonction `scanf()` est strictement interdit.**

```
int main()
```

Cette fonction, point d'entrée de notre programme, va uniquement initialiser l'aléatoire et appeler les fonctions nécessaires au bon déroulement du programme. Plus le `main()` est court, plus le code est clair ! Vous l'enrichirez au fur et à mesure de l'évolution de votre programme afin de le rendre facile à utiliser.

```
void compteAREbours (int duree)
```

Cette fonction permet d'afficher un compte à rebours sur `duree` secondes. Pour cela, vous aurez besoin d'utiliser une fonction permettant de mesurer le temps. Vous connaissez déjà la fonction `int time()` mais celle-ci n'offre une précision qu'à la seconde. Si cette dernière est suffisante pour l'affichage d'un compte à rebours, elle ne le sera plus pour la mesure du temps de réaction qui doit être précis au centième de seconde.

Nous allons donc lui préférer la fonction `clock_t clock()` qui offre une précision beaucoup plus importante puisque qu'elle compte le nombre de tics d'horloge du processeur depuis le lancement de la machine. En divisant cette valeur par la constante `CLOCKS_PER_SEC` – qui comme son nom l'indique, nous donne le nombre de tics d'horloge par seconde – nous obtenons le nombre de secondes écoulées depuis l'allumage de l'ordinateur. L'objectif est alors d'appeler la fonction `clock()` dès le début de la fonction `compteAREbours()` afin d'obtenir un point de départ pour le compte à rebours.

```
double tempsAleatoire (int max)
```

Cette fonction permet de générer un temps d'attente aléatoire dans le programme, d'une durée supérieure à une seconde, mais inférieure à `max` secondes. Les fonctions décrites précédemment nous seront une nouvelle fois utiles. Le processus pour bloquer l'exécution d'un programme est simple, il suffit d'exécuter une boucle d'instructions vide durant le temps souhaité.

```
double deroulerJeu (int duree, int max)
```

Cette fonction nous permet de dérouler le programme mesurant le temps de réaction. Elle affiche à l'écran la série de symbole représentant le temps de réaction écoulé. Durant cet affichage, l'appel à la fonction `int kbhit()` (fournie) nous permettra de vérifier si une touche clavier a été saisie par l'utilisateur. La fonction `kbhit()` renvoie la valeur 1 si une touche du clavier est appuyée, 0 sinon. Lorsque l'appui sur une touche est détecté, le programme doit arrêter d'afficher les symboles et retourner le temps de réaction de l'utilisateur.

La fonction `deroulerJeu` doit donc appeler les fonctions rédigées ci-dessus (sauf le `main`). Elle prend en paramètres l'entier `duree` correspondant au compte à rebours et l'entier `max` correspondant au temps d'attente maximal pour mesurer le temps de réaction.

Votre programme complet doit être simple et amusant à utiliser. Vous pouvez l'améliorer à volonté. Par exemple, après une première tentative, vous pouvez demander à l'utilisateur s'il veut rejouer. Vous pouvez également afficher un message différent suivant le temps de réaction mesuré. Soyez créatifs !