

# TD n°1

---

## Deviner le nombre

Il s'agit de programmer un jeu qui :

- fait choisir à l'ordinateur un nombre mystère compris entre 1 et 100 en utilisant un générateur pseudo-aléatoire ;
- demande à un utilisateur d'entrer un nombre ;
- compare la réponse de l'utilisateur avec le nombre mystère :
- si la réponse est égale au nombre mystère, affiche "gagné" ;
- si la réponse est strictement inférieure au nombre mystère, affiche "plus" ;
- si la réponse est strictement supérieure au nombre mystère, affiche "moins" ;
- recommence jusqu'à ce que l'utilisateur ait trouvé le nombre mystère ou ait fait 7 tentatives;
- affiche le score qui est 7 moins le nombre d'essais pour trouver le nombre mystère.
- Si l'utilisateur n'a pas trouvé le nombre mystère, il faut l'afficher.

### Question

Écrire une fonction qui permet de jouer au jeu tel que décrit ci-dessus.

**Indications :** Le code suivant permet de sélectionner un nombre aléatoire entre 1 et 100.

```
import 'dart:math'  
final aDeviner = 1 + Random().nextInt(100);
```

La saisie doit se faire dans le terminal, pas la console de débogage. Dans VS Code, menu Affichage, Palette de commandes..., Paramètres d'espace de travail, chercher "cliConsole" et choisir "Terminal"

Saisie au clavier et conversion :

```
import 'dart:io';  
String? chaine = stdin.readLineSync();  
// puis, si la chaîne n'est pas null  
int nb = int.parse(chaine);
```

Dans cet exercice, on manipule des chaînes de caractères ne contenant que les quatre caractères A, C, G et T, comme par exemple :

Ces chaînes de caractères décrivent des séquences ADN dont les éléments sont appelés bases.



## De l'ADN à l'ARN

### Question

retourne la chaîne

2 / 4

## Bases complémentaires

Les bases A et T sont dites complémentaires, ainsi que les bases G et C.

### Question

Écrivez une fonction `baseComplementaire` qui retourne la base complémentaire de celle passée en paramètre. Si le paramètre n'est pas une des bases A, C, G ou T, la fonction retourne la chaîne vide.

**Remarque :** une base est représentée par une chaîne de caractères de longueur 1.

Par exemple, l'appel

```
baseComplementaire('A')
```

retourne 'T' et l'appel

```
baseComplementaire('H')
```

retourne la chaîne de caractères vide.

## Séquences complémentaires

Dans les cellules vivantes, l'ADN est sous la forme de brin, c'est-à-dire 2 séquences ADN se font face. Une séquence est lue de gauche à droite et l'autre de droite à gauche. De plus, les bases complémentaires se font face l'une l'autre. Par exemple, les 2 séquences suivantes sont complémentaires inversées l'une par rapport à l'autre et peuvent ainsi former un double brin : GTACA et TGTAC.

### Question

Écrire une fonction `sequenceComplementaireInversee` qui retourne la séquence ADN complémentaire et inversée sous forme de chaîne de caractères de celle passée en paramètre. Vous utiliserez dans cette fonction la fonction `baseComplementaire`. On suppose que le paramètre est bien une séquence ADN. Par exemple, l'appel

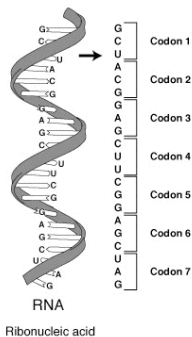
```
sequenceComplementaireInversee("ACTG")
```

retourne la chaîne de caractères

```
"CAGT"
```

## Codons

Le code génétique est écrit à partir des quatre bases de l'ARN. De plus, il est constitué de suites de trois bases appelées codons. Par exemple, les séquences suivantes sont des codons : ACG, GUA, CGC, GCU ... Le code génétique étant non chevauchant, la lecture des codons s'effectuera triplet par triplet.



## Question

Ecrivez une fonction nommée `nombreOccurrencesCodon` qui calcule le nombre de fois que le codon (représenté par une chaîne de caractères) passé en paramètre apparaît dans la séquence ARN également passée en paramètre. On suppose que la longueur de la séquence ARN est un multiple de 3. Par exemple, l'appel

```
nombreOccurrencesCodon('ACG', 'GCUACGGAGCUUCGGAGCACGUAG')
```

retourne 2 et l'appel

```
nombreOccurrencesCodon('UUC', 'AGUUCGACU')
```

retourne 0.