

# Contrôle continu de Java

Les questions sont classées par difficulté. Elles valent toutes le même nombre de points.

Les fichiers à compléter sont à télécharger sur <https://pro.aiakide.net/cours/TPA/>

Attention, la présentation (c'est à dire, essentiellement, l'indentation) est notée et vaut 10% des points. Il est fortement recommandé de tester vos actions et vos fonctions en les appelant depuis le programme principal ("main"), et vous pouvez laisser vos tests dans le fichier que vous rendez.

## 1. Date

Écrire une action *date* qui prend en argument trois entiers: un quantième de jour, un mois et une année, et qui affiche les affiche sous la forme "jj/mm/aaaa".

Par exemple `date(18, 10, 2018)` affiche "18/10/2018".

## 2. Mois

Écrire une fonction *estMois* qui prend en argument un entier, et retourne vrai si il est compris entre 1 et 12 (inclus), et faux dans le cas contraire.

## 3. MotDiese

Écrire une fonction *estMotdiese* qui prend en argument une chaîne de caractères, et retourne vrai si il s'agit d'un mot-dièse (hashtag en anglais – c'est à dire un mot qui commence par le caractère "#").

Vous pourrez utiliser les fonctions *characterAtPos* et *stringEquals* qui sont déjà présentes dans le fichier `MotDiese.java`.

## 4. Encadre

Écrire une action *encadre* qui prend en argument une chaîne de caractères, et affiche une ligne de tirets de même longueur, la chaîne en question, puis à nouveau le même nombre de tirets.

Par exemple:

`encadre("Un Exemple")` affiche:

```
-----  
Un Exemple  
-----
```

Rappel : `s.length()` retourne la longueur de la chaîne `s`.

## 5. Frequence

Écrire une fonction *frequence* qui prend en argument deux chaînes de caractères: *texte* et *car*. La fonction retourne le nombre d'occurrences de *car* dans *texte*, ou bien -1 si *car* ne contient pas exactement un caractère.

Vous pourrez utiliser les fonctions *characterAtPos* et *stringEquals* qui sont déjà présentes dans le fichier *Frequence.java*.

Par exemple:

`frequence("Un texte", "e")` retourne 2.

`frequence("Un texte", "te")` retourne -1.

`frequence("Un texte", "")` retourne -1.

# Contrôle continu de Java

Les questions sont classées par difficulté. Elles valent toutes le même nombre de points.

Les fichiers à compléter sont à télécharger sur <https://pro.aiakide.net/cours/TPB/>

Attention, la présentation (c'est à dire, essentiellement, l'indentation) est notée et vaut 10% des points. Il est fortement recommandé de tester vos actions et vos fonctions en les appelant depuis le programme principal ("main"), et vous pouvez laisser vos tests dans le fichier que vous rendez.

## 1. Temps

Écrire une action *temps* qui prend en argument deux entiers: une heure et un nombre de minutes, et qui affiche les affiche sous la forme "hh:mm".

Par exemple `temps(13, 37)` affiche "13:37".

## 2B. Heure

Écrire une fonction *estHeure* qui prend en argument un entier, et retourne vrai si il est compris entre 0 et 23 (inclus), et faux dans le cas contraire.

## 3B. Commande

Écrire une fonction *estCommande* qui prend en argument une chaîne de caractères, et retourne vrai si il s'agit d'une commande IRC (c'est à dire, si la chaîne commence par "!").

Vous pourrez utiliser les fonctions `charAtPos` et `stringEquals` qui sont déjà présentes dans le fichier `Commande.java`.

## 4. Encadre

Écrire une action *encadre* qui prend en argument une chaîne de caractères, et affiche une ligne de tirets de même longueur, la chaîne en question, puis à nouveau le même nombre de tirets.

Par exemple:

`encadre("Un Exemple")` affiche:

```
-----  
Un Exemple  
-----
```

Rappel : `s.length()` retourne la longueur de la chaîne `s`.

## 5. Frequence

Écrire une fonction *frequence* qui prend en argument deux chaînes de caractères: *texte* et *car*. La fonction retourne le nombre d'occurrences de *car* dans *texte*, ou bien -1 si *car* ne contient pas exactement un caractère.

Vous pourrez utiliser les fonctions *characterAtPos* et *stringEquals* qui sont déjà présentes dans le fichier *Frequence.java*.

Par exemple:

`frequence("Un texte", "e")` retourne 2.

`frequence("Un texte", "te")` retourne -1.

`frequence("Un texte", "")` retourne -1.