

Comme pour chaque feuille, commencer par créer un répertoire nommé ~/L1/IF1/TP3 qui sera le répertoire de travail pour ce TP.

Sans que cela soit rappelé systématiquement, chaque fonction doit être testée. Pour cela, il faut l'encapsuler dans une classe. En particulier, si un programme peut donner plusieurs résultats, il faudra le tester avec plusieurs jeux de données différents en entrée, au moins un par résultat possible.

1 Histoires simples et imbrications

- ▶ **Exercice 1** : Sans utiliser la classe `Math`, écrire une fonction `minimum` qui prend en arguments deux entiers et renvoie le plus petit des deux. Dans une même classe, écrire deux autres fonctions `minimum` qui prennent trois et quatre arguments entiers respectivement :
 - en utilisant des appels aux autres fonctions `minimum` de cette classe, ou
 - en utilisant des instructions conditionnelles avec ou sans imbrication.
- ▶ **Exercice 2** : Écrire une fonction `combien` qui prend en argument un nombre entier et renvoie la chaîne de caractères « moins que rien » si ce nombre est négatif, « aucun » si ce nombre vaut 0, « un » si ce nombre vaut 1, « deux » si ce nombre vaut 2 et « plusieurs » sinon.
- ▶ **Exercice 3** : Écrire une fonction `prix` qui prend en arguments le prix d'un kilo de pommes et le nombre de kilos achetés et renvoie le prix à payer, sachant que pour trois kilos ou plus achetés, le vendeur fera un rabais de 10% et que pour cinq kilos ou plus le rabais sera de 15%.
- ▶ **Exercice 4** (Partiel 2011, Exercice 3) : Au musée de Trevors, les tarifs d'entrée sont définis par ces règles :
 - pour les billets individuels : gratuité pour les enfants d'au plus 3 ans, tarif réduit (3€) pour les enfants entre 4 et 10 ans et les personnes âgées à partir de 65 ans et tarif plein (5€) pour les autres.
 - pour les billets de groupes (au moins 8 personnes) : sans considération d'âge, 10% de réduction sur le tarif plein pour les groupes jusqu'à 15 personnes incluses, 20% au-delà.
 Écrire des fonctions `tarifIndividu` et `tarifGroupe` renvoyant le prix à payer à partir d'arguments pertinents.
- ▶ **Exercice 5** (Partiel 2012, Exercice 5) : Pour faire des beignets, il faut 400g de farine, deux oeufs, 33 cl de lait, de l'huile et de la levure. On suppose ici que l'on a toujours suffisamment d'huile et de levure.
 1. En utilisant une des fonctions `minimum` de l'exercice 1, écrire une fonction `recette` qui prend en arguments une quantité de farine (en g), un nombre d'oeufs, un volume de lait (en cl) et renvoie sous forme de chaîne de caractères la recette de la pâte à beignets dans les proportions optimales de sorte que l'on puisse préparer le plus de beignets possible en fonction des ingrédients disponibles. Naturellement, le nombre d'oeufs doit être entier.
 2. Écrire une fonction `main` qui demande les quantités disponibles de farine, d'oeufs et de lait, puis affiche la recette optimale du beignet.

```
$ java Beignet
Farine disponible (en g) : 500
Oeufs disponibles : 1
Lait disponible (en cl) : 120
Recette optimale du beignet:
    200.0g de farine
    1 oeuf(s)
    16.5cl de lait
    huile
    levure
```

```
$ java Beignet
Farine disponible (en g) : 1100
Oeufs disponibles : 7
Lait disponible (en cl) : 100
Recette optimale du beignet:
    1000.0g de farine
    5 oeuf(s)
    82.5cl de lait
    huile
    levure
```

3. Cette recette pour beignets peut être allégée en remplaçant en partie (ou en totalité) le lait par de la bière. Toujours dans la recherche d'une recette optimale, proposer une modification des fonctions `recette` et `main` précédentes. Donner alors un exemple d'exécution.

2 Histoires de dates

- **Exercice 6** : Dans la classe `Bissextile`, mettre en commentaire la méthode `estBissextile` de l'exercice 5 du TP2. En écrire une nouvelle version utilisant des conditionnelles et plus aucun des opérateurs booléens de négation, de conjonction ou de disjonction.

Une année est dite *séculaire* si ses deux chiffres de droite sont nuls. Écrire une fonction `proprieteAnnee` qui prend un argument entier et renvoie la chaîne de caractères donnant les propriétés séculaire et/ou bissextile de l'année correspondante.

- **Exercice 7** : Écrire une fonction `minimum` qui prend en arguments deux dates (données chacune par trois entiers et supposées valides) et renvoie une chaîne de caractères donnant la plus ancienne des deux.

- **Exercice 8** (Partiel 2012, Exercice 3) : Pour déterminer le jour de la semaine correspondant à une date donnée par trois entiers `jour`, `mois` et `annee`, on calcule les entiers `s` et `d` correspondant aux deux chiffres de gauche et aux deux chiffres de droite de l'entier `annee`. On calcule ensuite le reste `r` de la division de `mois + 9` par 12 et l'entier `k` défini par la formule

$$\text{jour} + 366 + \left\lfloor \frac{2 + 13r}{5} \right\rfloor + \left\lfloor \frac{s}{4} \right\rfloor - 2s + \left\lfloor \frac{d}{4} \right\rfloor + d$$

dans laquelle $\left\lfloor \frac{p}{q} \right\rfloor$ désigne le quotient de la division de p par q . Pour janvier ou février d'une année non bissextile, on soustrait 1 à `k`. Pour janvier ou février d'une année bissextile, on soustrait 2 à `k`. Le jour de la semaine (de 0 pour dimanche à 6 pour samedi) est alors le reste de la division de `k` par 7.

1. Écrire une fonction `numeroJour` qui prend en argument une date donnée par des entiers `jour`, `mois`, `annee` (supposée valide) et renvoie le numéro du jour de la semaine.
2. En déduire une fonction `nomJour` qui prend en argument une date donnée par des entiers `jour`, `mois`, `annee` (supposée valide) et renvoie la chaîne de caractères donnant le nom du jour de la semaine.

- **Exercice 9** : Écrire une fonction `dateValide` qui prend en arguments trois entiers et teste si ces trois entiers peuvent correspondre respectivement au jour, au mois et à l'année d'une date. En déduire une fonction `lendemain` qui prend en arguments trois entiers correspondant à une date supposée valide et renvoie une chaîne de caractères donnant la date du lendemain.

- **Exercice 10** : Dans la classe `Trevors` de l'exercice 4, écrire une nouvelle fonction `tarifIndividuel` qui prend en arguments la date du jour (donnée par trois entiers) et la date de naissance de l'individu (`idem`).

- **Exercice 11** : La formule de l'exercice 8 est plutôt obscure. Proposer et implémenter un algorithme lumineux pour déterminer le nom ou le numéro du jour correspondant à une date donnée.

3 Histoires de noyades booléennes

- **Exercice 12** : Sur `didel`, récupérer le fichier `NoieLePoisson.java` en suivant les liens `Documents` et `liens`, `Feuilles_tp` puis `tp3`.

1. Que font les fonctions `noieLePoisson1` et `noieLePoisson2`? Les réécrire de telle sorte qu'elles ne contiennent chacune qu'une simple instruction `return`, la plus courte possible.
2. Que fait la fonction `noieLePoisson3`? La réécrire (petit à petit) de telle sorte qu'elle fasse le moins de lignes de code possible.