
Le but de ce TP est de se familiariser avec les **fonctions**. Vous ferez chaque exercice avec les squelettes fournis dans l'archive « INF1_TP3.zip » présent sur Moodle.

Exercice 1 : Énumérer l'alphabet

En Java, les caractères sont représentés par des entiers selon la norme ASCII¹. C'est une norme bien établie, et les différentes opérations sur les caractères sont définies en se basant sur leur code ASCII. Mais :

1. connaître la valeur des codes ASCII des caractères n'est pas l'objectif de INF1 ;
2. utiliser directement les codes ASCII pour désigner les caractères rend les programmes peu lisibles, alors que ce n'est pas toujours nécessaire (voire même rarement nécessaire).

Dans cet exercice, on exploite une propriété importante du code ASCII qui permet d'éviter de mentionner explicitement les valeurs des codes : les caractères alphabétiques (minuscules et majuscules) ont des codes **contigus**. Ainsi, étant donné un caractère alphabétique `ca`, le code du caractère "suivant" dans l'ordre alphabétique est `ca + 1`.

Il faut néanmoins veiller à une utilisation cohérente des types de données : `ca + 1` désigne le code entier du caractère suivant. Pour re-convertir ce code entier en un caractère, on peut utiliser une **conversion de type**² selon le modèle `(typeResultat) expression`. Ainsi, l'expression `(char) ('a' + 3)` désigne le caractère 'd'.³

En utilisant les explications précédentes, réalisez les exercices suivants.

1. Écrire une fonction qui affiche toutes les lettres de l'alphabet en minuscule à l'aide d'une boucle **for**. Tester votre fonction pour vérifier son bon fonctionnement.
2. Écrire une fonction qui affiche toutes les lettres de l'alphabet en majuscule, et dans l'ordre inverse (de Z à A), à l'aide d'une boucle **for**. Tester votre fonction pour vérifier son bon fonctionnement.

1. https://fr.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange

2. conversion de type : « *typecast* » en anglais

3. Attention aux priorités des opérateurs : la conversion de type est prioritaire sur l'addition. L'expression `(char)'a'+3` vaut 100, c'est-à-dire `97 + 3` (97 est la valeur du code ASCII du caractère 'a'), et non 'd'. Les parenthèses sont donc nécessaires dans `(char) ('a' + 3)` pour le résultat souhaité. Dans le cas général, il est conseillé d'utiliser des parenthèses pour ne pas dépendre des priorités des opérateurs.

Exercice 2 : Calcul d'impôts

En France, l'impôt sur le revenu est progressif. Les différentes tranches d'imposition sont :

- Fraction de revenu jusqu'à 10 064 € : 0 % .
- Fraction de revenu de 10 064 € à 25 659 € : 11 % .
- Fraction de revenu de 25 659 € à 73 369 € : 30 % .
- Fraction de revenu de 73 369 € à 157 805 € : 41 % .
- Fraction de revenu au-delà de 157 805 € : 45 % .

Par exemple, une personne gagnant 32000 € sera imposée de $10064 \times 0 + (25659 - 10064) \times 0.11 + (32000 - 25659) \times 0.3 = 3617,75$ €, soit 11,3 % de son salaire.

1. Écrire une fonction `impotMontant`, qui prend en entrée le salaire d'une personne et qui retourne le montant de son impôt.
2. Écrire une fonction `pourcentageImposition`, qui prend en entrée le salaire d'une personne et qui retourne son pourcentage d'imposition.

Pour les couples mariés/pacsés, on calcule l'impôt à payer sur la moyenne des deux salaires et on le multiplie par 2. Par exemple, si les conjoints gagnent respectivement 10000 € et 15000 €, ils auront $2 \times (10064 \times 0 + (12500 - 10064) \times 0.11) = 535,92$ € d'impôt à payer.

3. Écrire une fonction `gainMariage`, qui prend en entrée les deux salaires d'un couple et qui retourne la différence entre la somme des impôts de deux célibataires et ce qui est payé après mariage. Tester la fonction sur des exemples.

Exercice 3 : Cible

Le but de cet exercice est de réaliser une fonction qui prend en paramètre un entier n (compris entre 1 et 26 inclus) et qui affiche une cible de fléchettes formée de n carrés imbriqués. Les exemples suivants correspondent respectivement à $n = 3$ et $n = 4$.

NB : Pour cet exercice, tenez compte également des explications données en Exercice 1 à propos des énumérations des caractères.

<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>
<i>a</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>a</i>
<i>a</i>	<i>b</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>b</i>	<i>a</i>
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>c</i>	<i>c</i>	<i>b</i>	<i>a</i>
<i>a</i>	<i>b</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>b</i>	<i>a</i>
<i>a</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>a</i>
<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>

1. Écrire une fonction qui prend en entrée un entier i ainsi que n (avec $0 \leq i < n$) et qui affiche la ligne i de la cible.
2. Écrire la fonction qui affiche la cible.

Exercice 4 : Figure d'étoiles

Écrire un programme qui demande à l'utilisateur de saisir un entier h et qui affiche la figure suivante, où $3 * h$ est la hauteur de la figure (ici $h = 4$) :

```
      * * * *
    * * * * *
  * * * * * * *
* * * * * * * *
* * * * * * * *
* * * * * * * *
* * * * * * * *
  * * * * * *
    * * * * *
      * * * *
```