

## DM0 : Auto-test estival de vérification des acquis

Ce devoir maison, non noté, permettra une évaluation des compétences de base en algorithmique et programmation en sortie de lycée, qui sont des prérequis pour l'informatique de tronc commun en CPGE. Pour les exercices demandant d'écrire un programme, le langage Python est attendu mais il est possible d'utiliser du pseudo-code dans un premier temps. Lorsque les programmes calculent quelque chose, il faut être en mesure de savoir comment récupérer l'information calculée, afin de pouvoir s'en servir par ailleurs (attention à la différence entre une procédure qui affiche des messages, et une fonction qui va renvoyer une ou des valeurs).

A rendre par email à `mp2i@besson.link`, quand vous le souhaitez durant l'été, ou lors de la rentrée. Une photocopie scannée ou des photos bien propres suffisent, inutile de rédiger ça (sauf si vous connaissez déjà  $\text{\LaTeX}$ ).

**Exercice 1 :** Si la liste (de type `list` en Python) `toto` a été créée pour valoir `[3, 5, 0, 4, 2]`, quels sont les indices valides ? Et en particulier quel est l'indice de la valeur 2 ?

**Exercice 2 :** Déterminer laquelle des quatre propositions suivantes permet de créer une fonction `carre` permettant d'obtenir la valeur 1800 en écrivant dans la console `carre(42) + 36` :

<code>def carre(n):</code>	<code>def carre(n):</code>	<code>def carre(n):</code>	<code>def carre(n):</code>
<code>    return (n * n)</code>	<code>    disp(n * n)</code>	<code>    print(n * n)</code>	<code>    echo(n * n)</code>

**Exercice 3 :** Écrire un programme permettant de calculer le solde d'un compte où un euro a été déposé à l'ouverture avec des intérêts composés (c'est-à-dire sur le solde total) au taux permanent d'un pour cent par an, après deux mille ans.

**Exercice 4 :** Écrire un programme permettant de déterminer quel est le premier terme de la suite géométrique de terme général  $u_n = 3/(2^n)$  à être inférieur à un millième.

**Exercice 5 :** Écrire un programme permettant de construire une liste de six entiers aléatoires, tirés uniformément entre 1 et 6.

Rappel : pour obtenir un nombre aléatoire entre un seuil bas `a` et un seuil haut `b` (exceptionnellement en Python, tous deux sont inclus), on utilise `random.randint(a, b)`, l'utilisation de cette fonction nécessitant d'écrire auparavant (par principe en tout début) la ligne `import random`.

**Exercice 6 :** Écrire une fonction calculant la somme des éléments d'une liste `L` passée en paramètre.

**Exercice 7 :** Écrire une fonction construisant à partir d'une liste de nombres passée en paramètre une nouvelle liste dont les éléments sont des 1 (respectivement -1, 0) si au même indice dans la liste en paramètre le nombre est strictement positif (respectivement strictement négatif, nul).