

## 5IPRO Exercices récapitulatifs

### 1. Conditions

- a. Réalisez le code nécessaire pour afficher une variable "âge" uniquement si elle est strictement supérieure à 18 et inférieure à 50
- b. Réalisez le code nécessaire pour afficher une variable "âge" uniquement si elle est strictement égale en valeur et en type à 30
- c. Réalisez le code nécessaire pour vérifier qu'une variable "âge" uniquement si elle est supérieure à une variable "ageMinimum". Dans ce cas affichez soit un message indiquant que l'âge requis n'est pas atteint, ou que l'âge requis est atteint en fonction du résultat
- d. Réalisez la condition suivante de 3 manières différentes "Si il a un age alors je le multiplie par 2, sinon je le divise par 3"

### 2. Boucles

- a. Bouclez sur les entiers de 100 à 0, affichez la valeur de l'entier seulement si le résultat du modulo entre l'entier et une variable "comparaison" est différent de 0
- b. Bouclez sur les entiers de 2 à 200, faites la somme des entiers qui sont des nombres premiers (divisible uniquement par 1 et eux-même)

### 3. Tableaux

- a. Créez et parcourez un tableau comprenant des valeurs de différents types (boolean, string, integer, float, ..)
- b. Créez et parcourez un tableau d'entier, du dernier élément au premier
- c. Créez et parcourez un tableau d'entier du premier élément au dernier en évitant les index impairs et ce sans utiliser de condition à l'intérieur de votre boucle
- d. Créez et parcourez un tableau d'entier du premier élément au dernier en évitant les index dont le modulo 4 est 0
- e. Créez et parcourez un tableau en sommant les éléments à leurs place  
ex : [2,5,8,2] => [2, 7 (5+2), 15 (5+2+8), 17 (...)]

- f. Pareil que le précédent, sauf que vous avez un tableau de tableaux, une fois la somme faites triez les par ordre décroissant  
ex :  
Tableau de départ : `[[5, 2, 8], [12, 24, 9], [3, 1, 8]]`  
Tableaux avec sommes => `[[5, 7, 15], [12, 36, 45], [3, 4, 12]]`  
Tableaux ordonnés par ordre décroissant => `[[12, 36, 45], [5, 7, 15], [3, 4, 12]]`

#### 4. Tris

- a. Améliorez le tri à bulle tel que vu précédemment : Si lors d'un tour entier il n'a pas dû effectuer de "swap" alors arrêtez l'algorithme
- b. Réalisez un tri (bulle / insertion / sélection) décroissant
- c. Permettez à l'utilisateur d'entrer des valeurs (nombres entiers) remplissant un tableau, une fois que l'utilisateur est satisfait de ses entrées, triez le tableau par ordre croissant
- d. Créez un tableau d'entiers avec au moins 10 entrées, définissez ensuite une variable "limite", triez les N derniers éléments du tableau (N étant la limite)  
Quel algorithme choisir et pourquoi ? Implémentez le code
- e. Même question que la précédente sauf qu'il faut trier les N premiers éléments