

Colle n°4

Tous vos fichiers sont à **renommer** comme d'habitude (en y ajoutant votre nom en majuscule) et à **déposer sur Moodle** dans la section **Informatique - Devoirs**, tout en bas dans 1ère session. Vous devrez terminer les exercices de cette colle et me remettre l'ensemble des fichiers corrigés sur Moodle au même endroit avant dimanche soir.

Pensez à toutes les bonnes pratiques de programmation : réfléchir sur papier, tester dès que cela est possible, méthode des petits pas, programmation défensive...

Si vous avez terminé en moins d'1h, venez me demander un exercice supplémentaire.

Exercice 1 (Récursivité en OCaml (HUBERT, GRELY, ESCUDER)).

Elles seront écrites en OCaml dans un fichier `colle_recurivite.ml`

1. Écrire une fonction `somme` qui additionne deux listes terme à terme. Si l'une des listes est plus longue, la liste somme a la taille de la liste la plus courte.
2. Écrire une fonction `concatene_miroir` qui prend deux listes et les concatène en ordre inverse. Par exemple, `concatene_miroir [2;3;4] [5;6;7;8]` renvoie `[8;7;6;5;4;3;2]` L'utilisation de la fonction `rev` du module `List` est interdite.
3. Écrire une fonction `récursive_binaire : int -> int list` qui renvoie le tableau de la représentation binaire d'un entier.
4. Écrire une fonction `dispo` qui renvoie le plus petit entier **naturel** (i.e positif) qui n'apparaît pas dans la liste passée en argument. La liste peut contenir des entiers quelconques, y compris négatifs, et n'est généralement pas triée.
5. On définit un nouveau type par :

```
type couleur = Rouge | Blanc | Bleu;;
```

- a. Écrire une fonction `modification` de type `couleur list -> couleur list * bool` qui :
 - considère les couleurs consécutives et permute ces couleurs si elles sont mal ordonnées (l'ordre correct est celui du drapeau hollandais : rouge, puis blanc et enfin bleu) ;
 - averti s'il y a eu au moins une permutation lors du parcours de toute la liste en renvoyant `true` (et `false` sinon)
- b. Écrire une fonction `tri_drapeau` qui restaure le drapeau hollandais à partir d'un argument de type `couleur list`, c'est-à-dire qui remet les couleurs dans l'ordre de celles du drapeau hollandais.

Exercice 2 (Statistiques en C (AIT-EL-AMIRI, HOSFORD, GIRAULT)).

Toutes les fonctions de l'exercice seront codées dans un fichier `statistiques.c`

Privilégiez les choix d'architecture, la modularité et le découpage de votre code. Lisez l'ensemble de l'énoncé et faites le squelette avant de foncer sur votre clavier.

Vous avez beaucoup de liberté : faites tous les choix qui vous sont utiles et /ou vous simplifient la vie. Et n'oubliez pas : méthode des petits pas.

Une série statistique est la donnée de deux tableaux :

Tableau de valeurs : c'est un tableau de **valeurs réelles**

Tableau de fréquences : c'est un tableau d'**entiers positifs** qui indique la fréquence d'apparition de chacune des valeurs du tableau de valeurs

Ainsi, le tableau de valeurs est nommé `tab_val` et le tableau de fréquences `tab_freq`, la *i*-ème valeur du tableau `tab_val` est associée à la *i*-ème fréquence stockée dans le tableau `tab_freq`.

Une série statistique peut être stockée dans un simple fichier texte en listant d'abord les valeurs du tableau de valeurs et en les séparant par des espaces. Puis, après une retour à la ligne, en listant les valeurs de fréquence associées.

1. Créer trois fichiers de tests avec des séries statistiques de taille raisonnable. Avec votre calculatrice scientifique, calculer la moyenne et l'écart type de chacune des séries choisies.
2. Écrire un code C qui lit une série statistique fournie par un utilisateur dans un fichier et la stocke dans des tableaux. Elle devra s'assurer que la série statistique lue est valide. L'utilisateur donnera le nom du fichier de la série statistique sur la ligne de commande. Réfléchissez quelques minutes sur papier pour architecturer votre code
3. Écrire une fonction `nombre_echantillons` qui compte le nombre d'échantillons (effectif) de la série statistique
4. Écrire une fonction `max` qui retourne la valeur maximale de la série statistique
5. Écrire une fonction `moyenne` qui retourne la moyenne de la série statistique
6. Écrire une fonction `ecart_type` qui retourne la moyenne de la série statistique. On pourra utiliser la fonction `sqrt` de la bibliothèque `math.h`. Dans ce cas, il faut utiliser l'option `-lm` pour *linker* (édition de liens) la librairie mathématique lors de la compilation.