

Le but de ce TP est de reprendre l'exercice sur les dates vu en TD3 et de continuer à manipuler des tableaux.

Exercice 1 : Dates

1.a] Écrire une fonction `anneeBissextile` qui prend en paramètre un entier `annee` et qui retourne un booléen indiquant si l'année `annee` est une année bissextile ou non.

1.b] Écrire une fonction `jourDansMois` qui prend en paramètres deux entiers `mois` et `annee` et retourne le nombre de jour qu'il y a dans le mois (en tenant compte du fait que `annee` puisse être bissextile).

1.c] Écrire une fonction `dateValide` qui prend en paramètres trois entiers `jour`, `mois`, `annee` et qui retourne un booléen si oui ou non la date `jour/mois/annee` est valide ou non.

1.d] Écrire une fonction `joursEntreDates` qui prend en paramètres six entiers `jour1`, `mois1`, `annee1` et `jour2`, `mois2`, `annee2` et qui retourne le nombre de jours qui se sont écoulés entre `jour1/mois1/annee1` et `jour2/mois2/annee2`.

Exercice 2 : Histogramme - Tableaux de notes

Dans cet exercice, on manipule des tableaux d'entiers de longueur quelconque qui contiennent des notes comprises entre 0 et 20. Par exemple,

```
int[] notes = {6, 11, 13, 16, 12, 8, 9, 7, 12, 9, 8, 10, 14,
11, 9, 1, 6, 4, 19, 17, 12, 13, 4, 0, 5, 9, 10, 11};
```

Le but de cet exercice est de calculer et d'afficher un histogramme des notes. Un histogramme est un nouveau tableau donnant le nombre d'occurrences de chaque valeur possible, c'est-à-dire ici un histogramme est un tableau de 21 entiers dont la k -ième case contient le nombre de fois où la note k a été donnée.

2.a] Écrire une fonction `calculHistogramme` qui prend en paramètre un tableau `tab` et qui calcule et retourne l'histogramme associé. Par exemple, `calculHistogramme(notes)` retourne le tableau `{1, 1, 0, 0, 2, 1, 2, 1, 2, 4, 2, 3, 3, 2, 1, 0, 1, 1, 0, 1, 0}`.

2.b] Écrire une fonction `afficheHistoHorizontal` qui prend un histogramme `tab` en paramètre et l'affiche par lignes. Par exemple, si on utilise cette fonction sur l'histogramme précédent, on obtient l'affichage ci-contre.

2.c] (Challenge) Écrire une fonction `afficheHistoVertical` qui prend un histogramme `tab` en paramètre et l'affiche par colonnes. Par exemple, si on utilise cette fonction sur l'histogramme précédent, on obtient l'affichage suivant

```

                *
              *
            *   *   *
          * * * * *
        * * * * *
      * * * * *
    * * * * *
  * * * * *
* * * * *
0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

```

Exercice 3 : Polynômes

3.a] Écrire une fonction `affichePoly` qui prend en paramètre un tableau d'entiers `poly` et qui l'affiche sous le format d'un polynôme en rajoutant les puissances de `x`. Par exemple le tableau suivant `int[] f = {5, 7, 3, 2}`; sera affiché de la manière suivante $5 + 7x + 3x^2 + 2x^3$.

3.b] Écrire une fonction `evaluatePoly` qui prend en paramètres un tableau `poly` et un entier `y` et retourne l'évaluation du polynôme représenté par le tableau `poly` en la valeur `y`.

3.c] Écrire une fonction `additionPoly` qui prend en paramètres deux tableaux `poly1` et `poly2` représentant deux polynômes de degrés quelconques et qui retourne un nouveau tableau qui représente l'addition des deux polynômes passés en paramètres.

3.d] (Challenge) Écrire une fonction `derivee` qui prend en paramètre un tableau `poly` et qui retourne un nouveau tableau qui correspond à la dérivée de ce polynôme. On rappelle que la dérivée de cx^{n+1} est $c(n+1)x^n$.

3.e] (Challenge) Écrire une fonction `multiplicationPoly` qui prend en paramètres deux tableaux `poly1` et `poly2` représentant deux polynômes de degrés quelconques et qui retourne un nouveau tableau qui représente la multiplication des deux polynômes passés en paramètres. Indication : on peut commencer par écrire deux fonctions auxiliaires, `multiplicationScalaire` et `multiplicationPuissanceX` qui prennent en paramètres un tableau `poly` et un entier `a` et qui respectivement multiplie chaque coefficient par `a` et multiplie le polynôme par x^a .