

# BASES DE L'ALGORITHMIQUE

## CODING SPRINT N° 3

**Modalités :** Ce coding sprint, assorti d'un bonus pouvant aller jusqu'à 0,5 points sur la moyenne de l'examen TP 2, est facultatif.

Si vous souhaitez y participer, vous devez écrire vos programmes entre vendredi 13 octobre 2023 à 17h30 et lundi 16 octobre à 8h.

---

Vous allez écrire un programme qui permet, au choix de l'utilisateur :

1. de connaître le jour de la semaine correspondant à n'importe quelle date passée ou future,
2. de connaître le nombre de jours qui séparent deux dates quelconques,
3. de savoir depuis combien de jours il est né.

Remarque : La structure de votre programme comprendra de nombreuses fonctions. Cela permet de le rendre plus simple à lire et à analyser. Les fonctions étant faites pour être utilisées, vous ne devrez pas hésiter, lorsque cela s'y prête, à appeler une fonction depuis une autre fonction.

## 1 Quelques fonctions utiles

### 1.1 Années bissextiles

La plupart de années, le mois de février compte 28 jours. Lors des années dites *bissextiles* février compte 29 jours.

Pour savoir si une année est bissextile, on applique la règle suivante :

1. Les années multiples de 4 sont bissextiles.

Mais il y a une exception à cette règle :

2. Les années multiples de 100 ne sont pas bissextiles.

Mais il y a une exception à cette exception :

3. Les années multiples de 400 sont quand même bissextiles.

- Écrivez une fonction nommée `annee_bissextile` qui prend en entrée une année et qui retourne un booléen indiquant si cette année est bissextile ou ne l'est pas.

Par exemple, votre fonction doit retourner `False` pour l'année 2007, `True` pour l'année 2008, `False` pour l'année 1900 et `True` pour l'année 2000.

## 1.2 Nombre de jours dans un mois donné

- Écrivez une fonction nommée `nb_jour_mois` qui prend en entrée deux entiers (un mois et une année), et qui retourne le nombre de jours dans ce mois cette année-là.

Par exemple si le mois est égal à 12 et l'année à 2007, alors la fonction doit retourner 31, car décembre 2007 a 31 jours.

Si le mois est égal à 2 et l'année à 2008, alors la fonction doit retourner 29, car février 2008 a 29 jours.

## 1.3 Validité d'une date

- Écrivez une fonction nommée `date_valide` qui prend en entrée trois entiers représentant une date (un jour, un mois, une année), et qui retourne un booléen indiquant si cette date est valide ou non.

Par exemple la fonction devra retourner `False` si les 3 entiers sont 4, 14 et 2007. Elle devra retourner `True` si les trois entiers sont 29, 2, 2008. Enfin, elle devra retourner `False` si ce sont 29, 2, 2009 (car 2009 n'est pas bissextile).

## 1.4 Chaîne représentant une date

- Écrivez une fonction nommée `chaîne_date` qui prend en entrée trois entiers représentant une date valide, et qui retourne une chaîne de caractères représentant cette date.

Par exemple, si la fonction reçoit les entiers 29, 2 et 2008 elle devra retourner la chaîne "29 février 2008".

Autre exemple, si elle reçoit les entiers 1, 3 et 2008 elle devra retourner la chaîne "1er mars 2008".

Autrement dit votre fonction doit utiliser les textes des noms de mois, et afficher "1er" au lieu de "1" dans le cas où le jour est égal à 1.

Vous utiliserez cette fonction lorsque vous afficherez une date pour rendre le texte affiché plus agréable à lire.

## 2 Nombre de jours séparant deux dates

La fonction essentielle de votre programme devra déterminer le nombre de jours séparant deux dates reçues en paramètres.

Pour réaliser ce calcul, vous allez additionner :

- le nombre de jours séparant la date la plus ancienne et le 1<sup>er</sup> janvier de l'année suivante,
- les nombres de jours que comptent chaque année complète entre les deux dates,

- le nombre de jours séparant la date la plus récente et le 1<sup>er</sup> janvier de cette année-là.

Par exemple, si les deux dates sont le 25 novembre 2006 et le 12 janvier 2009, il faut additionner :

- **37** jours qui séparent le 25 novembre 2006 du 1<sup>er</sup> janvier 2007,
- **365** jours pour l'année 2007, et **366** jours pour l'année 2008,
- **11** jours qui séparent le 1<sup>er</sup> janvier 2009 du 12 janvier 2009.

Ce qui donne un total de 779 jours entre les deux dates.

Les différentes étapes de ce calcul inspirent le découpage suivant en plusieurs fonctions.

## 2.1 Nombre de jours dans une année

- Écrivez une fonction nommée `nb_jour_annee` qui prend en entrée une année et qui retourne le nombre de jours qu'il y a dans une année complète.

Par exemple, si la fonction reçoit l'année 2007 elle devra retourner 365. Si elle reçoit 2008 elle devra retourner 366.

## 2.2 Nombre de jours entre une date et la fin de l'année

- Écrivez une fonction nommée `nb_jour_fin_annee` qui prend en entrée trois entiers représentant une date valide et qui retourne le nombre de jours qui s'écoulent entre cette date et le 1<sup>er</sup> janvier de l'année suivante.

Par exemple, si la fonction reçoit 25, 11 et 2006 elle devra retourner 37.

## 2.3 Nombre de jours entre le début de l'année et une date

- Écrivez une fonction nommée `nb_jour_debut_annee` qui prend en entrée trois entiers représentant une date valide et qui retourne le nombre de jours qui s'écoulent entre le 1<sup>er</sup> janvier de cette année et cette date.

Par exemple, si la fonction reçoit 12, 1 et 2009 elle devra retourner 11.

## 2.4 Nombre de jours séparant deux dates

Nous avons maintenant à peu près ce qu'il faut pour compter le nombre de jours séparant deux dates.

- Écrivez une fonction nommée `nb_jour_entre_dates` qui prend en entrée six entiers représentant deux dates valides et qui retourne le nombre de jours qui séparent ces deux dates.

Par exemple, si la fonction reçoit 25, 11, 2006, 12, 1 et 2009 elle devra retourner 779.

Attention : Vous devrez gérer correctement (c'est-à-dire différemment de l'exemple donné en début de section 2) le cas où les deux dates sont dans la même année.

Remarque : Votre fonction devra vérifier que la première date est antérieure à la deuxième. Si c'est le cas elle retournera un entier positif. Dans le cas contraire elle retournera son opposé.

### 3 Le programme

- Écrivez un programme qui propose à l'utilisateur un menu lui permettant de choisir parmi 4 choix possibles :
  1. Calcul du jour de la semaine correspondant à une date.
  2. Calcul du nombre de jours séparant deux dates.
  3. Calcul du nombre de jours écoulés depuis sa naissance.
  4. Sortie du programme.

Ce menu sera affiché en boucle, et le programme ne sortira de cette boucle que lorsque l'utilisateur aura saisi le choix 4.

Chaque fois que vous demanderez à l'utilisateur de saisir une date (par les saisies successives de trois entiers), vous re-demanderez la saisie en boucle tant que la date n'est pas valide.

Les choix 4 et 2 ne présentent pas de difficulté particulière.

Pour le choix 1, vous vous baserez sur une date de référence dont vous connaissez le jour de la semaine

Par exemple vous pourrez calculer le nombre de jours séparant la date saisie et le 21 juillet 1969<sup>1</sup>. Puis, sachant que ce jour est un lundi, vous utiliserez un calcul de modulo pour en déduire le jour de la semaine de la date saisie.

Pour le choix 3, votre programme aura besoin de savoir quelle est la date du jour. Voici comment procéder :

1. Vous importez le module `datetime` via l'instruction `import datetime` placée en début du programme.
2. Vous obtenez la date du jour et l'affectez à une variable ainsi :  
`aujourd_hui = datetime.date.today()`
3. Vous pouvez alors accéder au jour, au mois et à l'année de la date du jour, avec respectivement `aujourd_hui.day`, `aujourd_hui.month` et `aujourd_hui.year`.

---

1. C'est le jour où pour la première fois un humain a marché sur la lune.

## 4 Et avant 1582 ?

Notre calendrier actuel s'appelle le calendrier *grégorien*. Il a été mis en place en 1582 à la demande du pape Grégoire XIII. Auparavant, c'est le calendrier *julien* (introduit par Jules César) qui était en vigueur.

Pendant la période du calendrier julien, les années bissextiles n'étaient pas définies de la même manière : seules les années multiples de 4 étaient considérées bissextiles.

Or, ne compter que les années multiples de 4 comme étant bissextiles a occasionné, avec le temps, un décalage de plus en plus grand entre le calendrier julien et le retour du soleil au point de l'équinoxe de printemps. Le décalage a ainsi atteint plus de 10 jours lorsqu'il fut décidé en 1582 de changer de calendrier pour passer au calendrier grégorien.

Le passage du calendrier julien au calendrier grégorien a occasionné deux changements :

- Un rattrapage des 10 jours de retard pris vis-à-vis de l'équinoxe du printemps. Il a ainsi été décidé que le lendemain du jeudi 4 octobre 1582 serait le vendredi 15 octobre 1582. Les dix dates intermédiaires n'ont jamais existé.
- Une modification de la définition des années bissextiles. La définition plus complexe adoptée par le calendrier grégorien (celle donnée à la section 1.1) permet, si ce n'est d'annuler, du moins de réduire significativement le rythme auquel le nouveau calendrier se décale vis-à-vis du soleil.

### 4.1 Amélioration de votre programme

En l'état actuel votre programme ne tient pas compte du changement de calendrier. En conséquence, il n'est utilisable que pour la période postérieure au 15 octobre 1582. Afin de le rendre compatible avec la période antérieure du calendrier julien, il est nécessaire de prévoir les changements suivants :

- La fonction permettant de savoir si une année est bissextile ou non doit être modifiée pour tenir compte des deux définitions différentes selon que l'année se situe dans la période du calendrier julien ou du calendrier grégorien.
- La fonction indiquant si une date est valide ou non doit tenir compte du fait que les dates allant du 5 au 14 octobre 1582 n'ont jamais existé.
- La fonction calculant le nombre de jours séparant deux dates doit compter 10 jours en moins si les deux dates sont de part et d'autre du changement de calendrier.