

Exercice 1 : Construire un programme permettant lorsqu'on lui fournit une date sous la forme numérique (3/2/5 où 3 indique le n° du jour de la semaine lundi=1, dimanche=7; le deuxième chiffre 2 indique le jour, enfin le troisième chiffre 5 indique le n° du mois) la convertit en clair (3/2/5 est converti en : mercredi 2 mai).

Exercice 2 : Ecrire un programme C# implémentant l'algorithme du tri à bulles :

```

Algorithme Tri_a_Bulles
local: i, j, n, temp ∈ Entiers naturels
Entrée - Sortie : Tab ∈ Tableau d'Entiers naturels de 1 à n éléments
début
pour i de n jusqu'à 1 faire // recommence une sous-suite (a1, a2, ..., ai)
pour j de 2 jusqu'à i faire // échange des couples non classés de la sous-suite
si Tab[j-1] > Tab[j] alors // aj-1 et aj non ordonnés
temp ← Tab[j-1];
Tab[j-1] ← Tab[j];
Tab[j] ← temp // on échange les positions de aj-1 et aj
Fsi
fpour
fpour
Fin Tri_a_Bulles

```

Exercice 2 : créez un programme qui demande à l'utilisateur d'entrer son année de naissance puis qui calcule, à partir de l'année courante, l'âge de la personne en question et l'affiche. Votre fonction Main ne fait aucune des tâches citées ci-dessus. Dans la classe effectuant tout le travail, vous séparerez chacune des actions exigées par cet exercice dans une méthode différente (il y en a donc 3 en tout). Au niveau de la gestion des exceptions, on vérifiera juste que l'année saisie est de type `int`.

Hint : pour connaître l'année actuelle lorsque le programme est exécuté, utilisez `DateTime.Today.Year` qui renvoie un entier.

Exercice 3 : Cette calculatrice, dont le menu correspond à l'image ci-dessous, devra faire en sorte que :

```

A = 3 ; B = 5
----- MENU PRINCIPAL -----
| 1 - Modifier A et B |
| 2 - Additionner A et B |
| 3 - Soustraire A et B |
| 4 - Multiplier A et B |
| 5 - Diviser(troncature) A et B |
| 6 - Diviser(reste) A et B |
| 7 - Quitter |
-----
Votre choix : _

```

- A et B soient deux attributs de type entier
- chaque choix possible soit traité et affiché dans une méthode différente
- la fonction Main n'effectue aucune des tâches du menu, ni même ne l'affiche
- l'affichage du menu est géré par une méthode qui renvoie un entier correspondant au choix de l'utilisateur
- avant d'afficher le menu, on nettoie la console (on la vide : on utilisera la méthode `clear` de la classe `Console`)
- l'utilisateur doit avoir le temps de pouvoir lire les résultats, qu'ils soient bons ou mauvais.

Exercice 4 : En utilisant tout ce qui a été vu, créez un programme répondant aux contraintes suivantes :

- Une classe doit lancer le programme, elle ne fait rien d'autre.
- Une classe s'occupe de répartir les tâches entre affichage et traitement des données.
- Une classe contient les données

Ce programme permet à l'utilisateur d'entrer le nom d'un album de musique ainsi que son artiste.

Un menu lui proposera :

- (1) d'entrer les données
- (2) d'afficher les données
- (q) de quitter

Vous devrez utiliser : au moins un constructeur, au moins une propriété et au moins une méthode. Vos attributs sont privés. N'hésitez pas à considérer les classes comme des types normaux et à les utiliser comme attributs d'autres classes. Les données ne sont pas stockées.

Exercice 5 : Créez un programme qui décompose un nombre entré par l'utilisateur en un produit de nombres premiers. Une propriété devra renvoyer le tableau d'entiers en question. Ce tableau aura été rempli par une méthode se trouvant dans la même classe. Ce programme aura l'aspect suivant :

- l'utilisateur entre un nombre (évités d'en entrer un trop grand du style à 100 chiffres)

- la factorisation en nombres premiers de son nombre est affichée sous la forme $\text{monNombre} = 2 * 3 * 5 * 7 * \dots * N$ où N est le plus grand nombre premier de la multiplication.

- **Attention à la gestion de la taille du tableau de facteurs.**

Aide: utilisez une boucle *for* et/ou *while* commençant à 2 et servez-vous en pour effectuer la division euclidienne ($\% = \text{modulo}$) du nombre. Si celle-ci renvoie 0, le nombre est divisible par le module en question et vous pouvez stocker le module et diviser le nombre par celui-ci. N'oubliez pas de vérifier à nouveau que le nombre n'est plus divisible par ce même module avant de l'incrémenter.

Exercice 6 : En vous basant sur l'exercice 4, réécrire ce programme en y ajoutant quelques fonctions :

- lorsque l'utilisateur lance le programme, celui-ci lui demande combien d'albums il compte ajouter. L'utilisateur doit entrer un nombre qui servira à déterminer la taille du tableau contenant les données.
- le menu devra proposer deux nouveaux choix : celui de pouvoir accéder directement aux informations d'un album par l'intermédiaire de son numéro. Les données de cet album pourront être soit modifiées, soit simplement affichées (ce qui nous fait bien 2 items de plus dans notre menu).
- lorsque l'utilisateur entre les données (lien (1) de l'ancien menu), il devra entrer toutes les données et remplir donc intégralement le tableau ; il en va de même lorsqu'il voudra les afficher (lien (2)).