

Les variables

1. Trouvez une méthode permettant d'inverser le contenu de deux variables (du même type évidemment). Si $A = 5$ et $B = 7$, après ce traitement, $A = 7$ et $B = 5$.

Les opérateurs arithmétiques

1. Déterminez la valeur des variables de ce pseudo-code.

```
Variable A, B, C, D, E : Entier
Debut
  A ← 8 MOD 3
  B ← 4 + A
  C ← B * A
  D ← (C - A) * B
  E ← ((A + 7) * (D DIV A)) * 0
Fin
```

2. Vérifiez vos réponses en utilisant C#.
3. Bonus : Imaginez une méthode permettant d'inverser le contenu d'une variable entière SANS utiliser une variable temporaire.
4. Réaliser un algorithme convertisseur de seconde. Ce dernier reçoit un nombre de secondes et détermine le nombre de jours, heures, minutes et secondes auquel elles correspondent.

Exemple : 4561 secondes correspondent à 0 jour 1 heure 16 minutes et 1 seconde.

Réfléchissez à la méthode que nous devons utiliser.

Une fois l'algorithme réalisé, testez-le en C#.

Les opérateurs de comparaison et les opérateurs logiques

1. Considérons $A = 3$, $B = 9$, $C = \text{Faux}$, $D = \text{NON}(C)$. Donnez le résultat pour chacune de ces instructions :

| Notation pseudo-code | Notation C# |
|---|-----------------------------|
| 1. $(A > 8)$ | 1. $A != 3$ |
| 2. $(B = 9)$ | 2. $!(D) C$ |
| 3. $\text{NON}(A \neq 3)$ | 3. $((A + B) == 12) \&\& D$ |
| 4. $\text{NON}(C)$ | |
| 5. $(A < B) \text{ OU } C$ | |
| 6. $\text{NON}((A + B) \neq 12)$ | |
| 7. $((B = 5) \text{ OU } ((c > 10) \text{ ET } (a < 8)))$ | |
| 8. $((((b = 5) \text{ OU } ((c > 10) \text{ ET } (a < 8))) \text{ OU } (A < B) \text{ OU } C) \text{ ET } C)$ | |

Les structures conditionnelles

1. Année bissextile (Pseudo-Code + C#)

Réaliser un petit algorithme qui sur base d'une année donnée va déterminer s'il s'agit d'une année bissextile. Une année est bissextile si elle est divisible par 4, mais non divisible par 100. Ou si elle est divisible par 400.

2. Lanceur de balles de tennis (Pseudo-Code)

Réaliser l'algorithme d'un lanceur de balles de tennis. Ce lanceur possède deux états :

- prêt : permet de savoir si le tennisman est prêt. Il ne faut pas lancer de balles dans le cas contraire
- panierVide : permet de savoir s'il y a encore des balles disponibles

Le lanceur de balle possède l'opération « lancerBalle » qui, vous l'aurez compris, permet de lancer une balle.

3. Distributeur de boissons (Pseudo-Code)

Réaliser l'algorithme d'un distributeur de boissons. Ce dernier propose plusieurs boissons et l'utilisateur choisit celle qu'il désire en entrant le numéro correspondant. Ne pas oublier pas de vérifier s'il y a encore des boissons en stock.

4. Calculatrice (Pseudo-Code + C#)

Réaliser l'algorithme d'une calculatrice basique. L'utilisateur est invité à saisir un nombre, un opérateur, et un deuxième nombre. La calculatrice affiche ensuite le résultat.

5. Différence entre deux durées

À l'aide de l'algorithme convertisseur de seconde, réaliser un algorithme faisant la différence entre deux durées (exprimées en jours, heures, minutes, secondes).

Les structures itératives

1. À l'aide d'une boucle, afficher la table de multiplication par 2. Ensuite, coder votre algorithme en C#.
2. Reprenez l'algorithme du lanceur de balles de tennis et faites en sorte qu'il lance une balle tant que le stock n'est pas vide.
3. À l'aide de deux boucles, afficher les tables de multiplication de 1 à 9. Ensuite, coder votre algorithme en C#.
4. Réfléchissez à des idées d'amélioration de notre distributeur de boissons maintenant que nous connaissons les boucles.

5. Réalisez un système de connexion à l'aide d'un mot de passe. L'algorithme demande à l'utilisateur de saisir son mot de passe. Si ce dernier valide de bon mot de passe, on le salue. Par contre, si il fait une erreur trois fois de suite, un message lui signalera que son compte est bloqué et il ne pourra pas réessayer une quatrième fois

6. Réalisez un algorithme qui demande un nombre à l'utilisateur et affiche autant de ligne que le nombre spécifié par l'utilisateur.

Exemple : l'utilisateur a rentré le nombre 5 et l'algorithme affiche :

```
*
**
***
****
*****
```

7. Réalisez le jeu du Plus ou Moins.

L'ordinateur choisi un nombre aléatoirement entre 1 et 100. L'utilisateur est invité à entrer un nombre et l'algorithme nous répond "C'est plus" ou "C'est moins". Lorsqu'on a trouvé le bon nombre, l'algorithme affiche le nombre de tentatives effectuées pour trouver le résultat.

8. Ecrire un algorithme qui demande à l'utilisateur de taper 10 entiers et qui affiche le plus petit de ces entiers.
9. Algorithme demandant 3 nombres : nbRep, nbTiret, nbEspace. Ce dernier affiche à l'écran autant de tiret que la valeur de nbTiret, suivi d'autant d'espace que la valeur de nbEspace. Le tout autant de fois que la valeur de nbRep.

Exemple : si nbRep = 2, nbTiret = 1 et nbEspace = 3 le résultat est le suivant :

| | | | |
|--|---|---|--|
| | - | - | |
|--|---|---|--|

Les structures itératives (2)

1. À l'aide d'une boucle Faire... Jusqu'à, améliorez l'algorithme du distributeur de boissons pour qu'il demande au client s'il désire une autre boisson (jusqu'à ce qu'il n'en ait plus envie).
2. À l'aide d'une boucle Faire... TantQue, améliorez l'algorithme de la calculatrice afin qu'elle demande à l'utilisateur s'il veut faire un autre calcul (tant qu'il le désire).
3. À l'aide de la boucle de votre choix, réaliser un algorithme calculant le résultat de N^{10} . N étant un nombre saisi par l'utilisateur.
4. Reprenez l'exercice précédent et modifiez le pour que l'utilisateur entre également l'exposant qu'il désire calculer.

Les tableaux

1. Écrire un algorithme qui saisit 6 entiers et les stocke dans un tableau, puis affiche le contenu de ce tableau une fois qu'il est rempli.
2. BONUS : initialiser un tableau de 10 entiers avec les valeurs 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 à l'aide d'une boucle. Ensuite, à l'aide d'une boucle afficher la valeur de chaque cellule du tableau avec l'opération `Ecrire()`.
3. Écrire un algorithme demandant à l'utilisateur le nombre de joueurs (max 10 joueurs). Ensuite, l'algorithme doit demander à l'utilisateur le score de chaque joueur. Une fois ceci fini, il faut afficher la moyenne des scores. Faites de même en C#.
4. Inverser un tableau : soit un tableau T. Saisir ce tableau. Changer de place les éléments de ce tableau de façon à ce que le nouveau tableau soit une sorte de miroir de l'ancien et afficher le nouveau tableau.
5. À l'aide des boucles, réalisez un algorithme permettant de trier un tableau d'entier dans l'ordre croissant. Mettez-le ensuite en pratique avec le C#.
6. En considérant deux tableaux d'entiers (non triés), réalisez un algorithme qui place tous les éléments des deux tableaux dans un troisième. Ce dernier doit être trié une fois l'algorithme terminé. Notez que le tri doit être fait en même temps que la fusion des deux tableaux et pas après.
7. Réalisez un algorithme nous permettant de déplacer un pion dans un tableau de 10 éléments. Au début, le pion se trouve dans la première case du tableau. Nous pouvons ensuite le déplacer par la gauche (g), par la droite (d) ou de stopper l'algorithme (q).
8. Réalisez un algorithme permettant de rechercher une valeur dans un tableau. Si la valeur se trouve bien dans la tableau, nous affichons sa position.
9. Refaites l'algorithme qui demande à l'utilisateur de taper 10 entiers et qui affiche le plus petit de ces entiers mais cette fois-ci à l'aide d'un tableau et sans retenir le minimum lors de la saisie.