

## Exercice n°1 : Instructions d'entrée-sortie : Échange de valeur

Dans cet exercice, on se propose d'étudier les algorithmes suivants, en exécutant pas à pas chacune des instructions, et en regardant l'évolution du contenu de chacune des variables.

- a) Échange de deux variables entières, sans variable temporaire

```
Algorithme Échange de deux variables entières
{
    int variable_1, variable_2 ;

    // Saisie des valeurs
    std::cout << "Saisir la valeur de la première variable\n" ;
    std::cin >> variable_1 ;
    std::cout << "Saisir la valeur de la deuxième variable\n" ;
    std::cin >> variable_2 ;

    // Permutation
    variable_1 = variable_1 + variable_2
    variable_2 = variable_1 - variable_2
    variable_1 = variable_1 - variable_2

    // Affichage du résultat
    std::cout << "La valeur de la première variable est : " << variable_1 << "\n";
    std::cout << "La valeur de la deuxième variable est : " << variable_2 << "\n";
}
```

- b) Échange de deux variables entières, avec deux variables temporaires

```
Algorithme Échange de deux variables entières avec 2 variables temporaires
{
    int variable_1, variable_2, ancien_1, ancien_2 ;

    // Saisie des valeurs
    std::cout << "Saisir la valeur de la première variable\n" ;
    std::cin >> variable_1 ;
    std::cout << "Saisir la valeur de la deuxième variable\n" ;
    std::cin >> variable_2 ;

    // Permutation
    ancien_1 = variable_1 ;
    ancien_2 = variable_2 ;
    variable_1 = ancien_2 ;
    variable_2 = ancien_1 ;

    // Affichage du résultat
    std::cout << "La valeur de la première variable est : " << variable_1 << "\n";
    std::cout << "La valeur de la deuxième variable est : " << variable_2 << "\n";
}
```

c) Échange de deux variables entières, avec une variable temporaire

```
Algorithme Échange de deux variables entières avec 1 variable temporaire
{
    int variable_1, variable_2,ancien_1 ;

    // Saisie des valeurs
    std::cout << "Saisir la valeur de la première variable\n" ;
    std::cin >> variable_1 ;
    std::cout << "Saisir la valeur de la deuxième variable\n" ;
    std::cin >> variable_2 ;

    // Permutation
    ancien_1 = variable_1
    variable_1 = variable_2
    variable_2 = ancien_1

    // Affichage du résultat
    std::cout << "La valeur de la première variable est : " << variable_1 << "\n";
    std::cout << "La valeur de la deuxième variable est : " << variable_2 << "\n";
}
```

## Exercice n°2 : Instructions d'entrée-sortie : Vos premiers algorithmes

Pour les questions suivantes :

- Faites l'inventaire des données entrées et sorties ;
- Écrivez les expressions mathématiques (s'il y a lieu) permettant d'obtenir les résultats désirés ;
- Concevez un modèle d'écran ;
- Écrivez l'algorithme.

a) Votre premier algorithme :

- Calculez et affichez le produit de 3 nombres entiers.
- Quelle est la partie de l'algorithme précédent à modifier pour calculer et afficher le produit de 3 nombres décimaux ?

b) Curiosité

Calculer l'âge (en année) d'une personne dont on aura demandé le nom et l'année de naissance, puis afficher le texte suivant :

**xxx[nom]xxx vous avez 999 ans**

remarque : xxx[nom]xxx représente le nom et 999 représente l'âge de la personne.

c) Comme Pythagore

Calculez et affichez le périmètre et la surface d'un carré, d'un rectangle et d'un cercle (avec  $\pi=3.1416$ ).

d) Une pincée de mathématiques

Écrivez l'algorithme réalisant le calcul de la fonction  $y = a*x^2 + b*x + c$  en tenant compte du modèle d'écran suivant :

Fonction  $y = a*x*x + b*x + c$

Valeur de a ?

Valeur de b ?

Valeur de c ?

Valeur de x ?

Résultat y =

e) **Économisons**

Calculez et affichez les intérêts I, simples et non capitalisés après chaque année, au taux annuel T pour un capital C après un placement de N années.

f) **Courage**

Calculez et affichez la valeur du dernier terme L et de la somme S des N premiers termes d'une progression arithmétique de raison R et dont le premier terme est A. Les formules sont les suivantes :

$$L = A + (N-1)*R \text{ et } S = N*(2*A + (N-1)*R)/2$$

### **Exercice n°3 : La structure conditionnelle : Étude d'un algorithme**

- a) Simuler l'exécution de l'algorithme suivant en utilisant comme valeurs pour A, B et C : (5, 8, 9) puis (2, 6, 3) et enfin (3, 2, 1). En déduire le nom de l'algorithme.
- b) Déterminer le nombre d'affectations et de comparaisons engendrées par l'algorithme (une fourchette si ce nombre varie).

Algorithme Nom de l'algorithme à déterminer

```
{
    int A, B, C ,p, d, t ;

    // Lecture des variables
    std::cin >> A >>B >> C ;

    // Processus
    if (A < B)
    {
        if (C > B)
        {
            p = A ;
            d = B ;
            t = C ;
        }
        else
        {
            if (A < C)
            {
                p = A ;
                d = C ;
                t = B ;
            }
            else
            {
                p = C ;
                d = A ;
                t = B ;
            }
        }
    }
    else
    {
        if (C > A)
        {
            p = B ;
            d = A ;
            t = C ;
        }
        else
        {
            if (B < C)
            {
                p = B ;
                d = C ;
                t = A ;
            }
            else
            {
                p = C ;
                d = B ;
                t = A ;
            }
        }
    }

    // Affichage du resultat
    std::cout << p << d <<t << std::endl ;
}
```