

Exercice sur les tableaux et les fonctions

Par **Godefroy Borduas** pour le cours 420-C21-IN (*Hiver 2022*)

Exercice 1

À partir du code suivant, donnez le résultat de l'exécution.

```
#include <iostream>
using namespace std;

int n = 10, q = 2;

int foo(int);
void bar(int);

void main(void) {
    int q = 5;
    n = foo(q);
    cout << "Main : n = " << n << ", q = " << q << "\n";
    bar(2);
}

int foo(int p) {
    int q = p * n;
    cout << "Foo : n = " << n << ", q = " << q << "\n";
    return q;
}

void bar(int p) {
    cout << "Bar : n = " << n << ", q = " << q << ", p = " << p << "\n";
}
```

Exercice 2

Écrivez une fonction, sans argument ou valeur de retour, qui affiche le nombre d'appel réalisés à la fonction depuis le lancement du programme. Oubliez le prototype.

Exercice 3

Écrivez une fonction qui calcule le [plus grand diviseur commun](#) grâce à l'[algorithme d'Euclide](#).

Exercice 4

Écrivez une fonction qui calcul le [plus petit multiplicateur commun](#) à l'aide du [plus grand diviseur commun](#).

Exercice 5

Écrivez une fonction qui ajoute, par référence, un nombre entier p au nombre entier n . Cette fonction n'a pas de retour.

Exercice 6

Écrivez une fonction qui recherche, dans un tableau quelconque de nombre flottant, la plus grande valeur et qui retourne sa position.

Exercice 6

Écrivez une fonction qui recherche, dans un tableau quelconque d'entier, la plus petite valeur multiple de 3 et qui retourne sa position. Si le tableau ne contient pas de multiple de 3, elle doit retourner -1;

Exercice 7

Prenez la suite suivante :

$$R(x > 2) = R(x - 1) \times R(x - 2) - 2 \times R(x - 3)$$

$$R(2) = 2 \quad R(1) = 5 \quad R(0) = 1 \quad R(x < 0) = 0$$

Exercice 7.A

Écrivez une fonction qui calcule, récursivement, la suite précédente.

Exercice 7.B

Écrivez une fonction qui remplit un tableau d'entier avec N termes de la suite. Cette fonction doit commencer à l'indice 0 et utiliser la fonction précédente.

Exercice 7.C**

Récrivez les fonctions `R` et `TableauR` de façon à réutiliser les valeurs précédentes. Vous devez toujours utiliser deux fonctions. De plus, seule la fonction `TableauR` contient la taille du tableau.

Exercice 8

Écrivez une fonction qui reçoit deux tableaux d'entier `t1` et `t2`. Les deux tableaux ont la même taille. Dans cette fonction, vous devez recopier tous les éléments de `t1` qui sont un nombre impair et un multiple de 5 dans le tableau `t2`. Enfin, l'excédent du tableau `t2` qui ne contient pas de valeur copiée doit être initialisé à 0.

Exercice 9

Il existe plusieurs algorithmes de tri. L'algorithme de [tri rapide](#) est très simple à implémenter. Son pseudo-code est le suivant :

```
fonction partitionner(tableau T, entier premier, entier dernier, entier pivot)
    échanger T[pivot] et T[dernier] // échange le pivot avec le dernier du
    tableau , le pivot devient le dernier du tableau
    j := premier
```

```
    pour i de premier à dernier - 1 // la boucle se termine quand i = (dernier-1).
        si T[i] <= T[dernier] alors
            échanger T[i] et T[j]
            j := j + 1
    échanger T[dernier] et T[j]
    renvoyer j

fonction tri_rapide(tableau T, entier premier, entier dernier)
    si premier < dernier alors
        pivot := choix_pivot(premier, dernier)
        pivot := partitionner(T, premier, dernier, pivot)
        tri_rapide(T, premier, pivot-1)
        tri_rapide(T, pivot+1, dernier)
```

Le pivot est choisi aléatoirement dans l'intervalle. Implantez ces deux fonctions en C++ pour un tableau de nombre flottant.

Exercice 10

En mathématique, la médiane représente la tendance centrale dans une série de données. Écrivez une fonction **Médiane** qui retourne la valeur médiane pour un tableau de N nombres flottants.