

TP 2 SDA Les tableaux

Pour les booléens, on pourra utiliser au choix

- des macros (BOOL TRUE & FALSE)
- le type bool si on compile en C99 (gcc -std=c99 ... + #include<stdbool.h>)

Tableaux 1D (int)

- Ecrire les fonctions suivantes ainsi que les tests correspondants:
- INT est le type du tableau (pour distinguer valeur/indice)
- void print(INT array[], int size)
- void read(INT array[], int size)
- INT valeur_max(INT array[], int size)
- void swap_elt(INT array[], int size, int i, int j)
- int occurrences(INT array[], int size, INT el)
- int find_first(INT array[], int size, INT el)
- int find_last(INT array[], int size, INT el)
- void reverse(INT array[], int size)
- void rotate_right(INT array[], int size)
- void rotate_left(INT array[], int size)
- bool is_palindrome(INT array[], int size)

Remarque on travaille ici par effet de bord !

Implanter le drapeau hollandais

- on utilisera ici un tableau de char
- on utilisera rand() [et srand()] pour générer des nombres aléatoire
- générer un tableau (taille 30) contenant aléatoirement les caractères R W B
- implanter la solution linéaire en 1 passe
- afficher le résultat à chaque étape
- vérifier qu'on est bien en O(n)

variante à 2 ensembles de réels

- Le but est de trier un tableau de réels en mettant tous les négatifs avant les positifs
- générer un tableau contenant aléatoirement des réels entre -1 et 1
- implanter une solution linéaire en 1 passe
- afficher le résultat à chaque étape

Tableaux 2D (int)

- Ecrire les fonction suivante ainsi que les test correspondant:
- `void print2(INT array[][3], int size)` // size = nb_lignes ou nb_elts ?
- `void read2(INT array[][3], int size)`
- `INT valeur_max2(INT array[][3], int size)`
- `int occurrences2(INT array[][3], int size, int el)`
- `bool is_sudoku(int array[3][3])` (les éléments sont les nombres 1,...,9 sans répétition)

Matrices 3x3 réelles

- utiliser `typedef float Mat33[3][3];`
- `readMat(Mat33 M)`
- `setId(Mat33 M)` initialize à la matrice identité (diagonale à 1, le reste à 0)
 - faire une version optimisée (2 boucles, 1 passe et pas de if)
- `printMat(Mat33 M)`
- `somme(Mat33 A, Mat33 B, Mat33 ApB)`
- `produit(Mat33 A, Mat33 B, Mat33 AfB)`

Attention l'utilisation du typedef qui cache le tableau porte ici à confusion, car il devient difficile de deviner que l'on travaille avec effet de bord.