

Exercice 1 – Tasser un tableau

1. Écrire une fonction `elimine`, prenant en paramètres un entier `c`, un tableau `A` de type `int` et la taille `n` de ce tableau, qui élimine toutes les occurrences de `c` dans `A` en tassant les autres éléments. La fonction renverra également la nouvelle taille du tableau.
Par exemple, le tableau $\{1,9,4,6,9,9,6,9,3\}$ deviendra $\{1,4,6,6,3\}$ après élimination du chiffre 9.
2. Écrire une fonction `eliminedouble`, prenant en paramètres un tableau de type `int` et la taille de ce tableau, qui ne garde que les premières occurrences des éléments du tableau, en tassant les éléments restants. La fonction renverra également la nouvelle taille du tableau.
Par exemple, le tableau $\{1,9,4,3,9,4,4,1,6\}$ deviendra $\{1,9,4,3,6\}$.

Exercice 2 – Objectif π

On peut trouver un certain nombre de calculs qui convergent vers le nombre π .

Par exemple, le produit des $\frac{1}{1-\frac{1}{4 \times i^2}} = \frac{4 \times i^2}{4 \times i^2 - 1}$ pour i variant entre 1 et n converge vers $\frac{\pi}{2}$ quand n devient grand.

Calculer et afficher ce produit pour différentes valeurs de n .

Par exemple, pour $n = 3$, on doit afficher le résultat de $\frac{4}{4-1} \times \frac{4 \times 4}{4 \times 4 - 1} \times \frac{4 \times 9}{4 \times 9 - 1}$.

Exercice 3 – Carré magique

- Un *carré magique* est un carré divisé en cellules dans lesquelles les nombres entiers, à partir de 1, sont arrangés de telle manière que les sommes de chaque ligne, de chaque colonne et de chaque diagonale soient égales.

Exemple

4	9	2	
3	5	7	somme=15
8	1	6	

Plusieurs algorithmes permettent d'obtenir des carrés magiques d'ordre impair (le nombre d'éléments par côté est impair). Le plus simple est décrit ci-dessous :

Nous présentons une méthode de complexité $O(n^2)$ pour créer des carrés magiques d'ordre impair.

1. On commence par initialiser tous les éléments du tableau à zéro (i.e. toutes les cases sont libres).
2. Le chiffre 1 est placé dans la case au milieu de la dernière ligne du carré.
3. Une fois qu'une case a été remplie, on en choisit une autre en effectuant, par rapport à la case qui vient d'être remplie précédemment, deux mouvements successifs :
 - l'un horizontal vers la case de droite
 - l'autre vertical d'une case vers le haut

Si i est l'indice de ligne et j l'indice de colonne, ce déplacement s'obtient en effectuant $i = i+1$ et $j = j+1$. Deux cas peuvent alors se présenter :

- (a) La case atteinte est libre, on y place alors le nombre suivant.
 - (b) La case atteinte est déjà remplie, on en choisit une autre en effectuant un déplacement vertical vers le haut à partir de la case de départ.
4. Dès qu'un déplacement oblige à sortir du carré, on se place dans la case correspondante du bord opposé et on continue à effectuer la suite des déplacements.

Ecrivez un programme qui engendre un carré magique en utilisant cet algorithme.

- Écrire une procédure **initialiser** qui prend en paramètre une matrice carrée t et sa taille n , et qui l'initialise à 0 et place le chiffre 1 au milieu de la dernière ligne du carré.
- Écrire une procédure **Afficher** qui prend en paramètre une matrice carrée t et sa taille n , et qui l'affiche
- Écrire une fonction **Verifier** qui prend en paramètre une matrice carrée t et sa taille n et qui vérifie si le carré est bien magique.
- Écrire une fonction **Placer** qui prend comme paramètres une matrice carrée t , sa taille n et deux entiers i, j représentant la case à remplir