

Une feuille A4 autorisée – Appareils électroniques interdits

Exercice 1 (4 points, environ 15 minutes)

On veut savoir combien d'éléments dans un tableau sont compris entre deux bornes. Par exemple, le nombre d'éléments compris entre 8 et 12 dans le tableau ci-contre est 3. Pour cela :

7	12	4	11	17	3	2	9
---	----	---	----	----	---	---	---

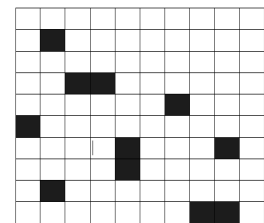
- écrire la fonction booléenne `dansIntervalle` qui étant donné un entier x et deux bornes min et max renvoie `true` si $x \geq min$ et $x \leq max$ et `false` sinon ;
- utiliser cette fonction dans l'écriture de la fonction `nbElemDansIntervalle` qui renvoie la valeur recherchée, étant donné un tableau et deux bornes.

Exercice 2 (2 points, environ 10 minutes)

Dans le tri-bulle, si le tableau devient ordonné avant le dernier passage, l'algorithme continue à envisager inutilement des échanges qui ne se feront pas. Or, il existe un moyen simple de savoir dans cet algorithme que le tableau est devenu complètement ordonné. Indiquer en français ce qu'il faudrait modifier pour éviter ces passages inutiles.

Exercice 3 (4 points, environ 20 minutes)

On dispose d'une grille représentée par un tableau `t` à deux dimensions quelconques de booléens. Les cases noires sont représentées par la valeur `true` et les cases blanches par la valeur `false`. Ce tableau est déjà initialisé.



- Écrire la fonction `proportionTrue` qui étant donné le tableau renvoie la proportion (entre 0 et 1) de cases noires.
- Écrire la fonction `ligneBlanche` qui renvoie `true` s'il existe une ligne sans aucune case noire et `false` sinon (ne pas parcourir systématiquement toutes les lignes !).

Exercice 4 (3 points, environ 15 minutes)

Lorsque l'on veut lire un entier avec `nextInt()`, il arrive parfois que l'utilisateur se trompe et saisisse des caractères qui ne constituent pas un entier (par exemple, `53R1` ou `7-`). Pour éviter ce problème, on peut lire une chaîne de caractères et vérifier qu'elle correspond à un entier avant de la convertir. Écrire la fonction booléenne `vérificationEntier` qui retourne `true` si le paramètre de type `String` qui lui est passé correspond bien à un entier : une suite de chiffres avec éventuellement le caractère '-' au début. Ainsi, `verificationEntier("-17")` ou `verificationEntier("17608")` doivent retourner `true` alors que `verificationEntier("34b5")` ou `verificationEntier("--67")` doivent retourner `false`.

Exercice 5 (7 points, environ 30 minutes)

On veut créer une copie d'un fichier contenant un mot par ligne, en recodant chaque mot. Chaque mot est remplacé par un décalage de ses lettres de 2 crans à droite. Par exemple, `"LAMPADAIRE"` devient `"RELAMPADAI"`. Les mot d'une seule lettre sont recopiés à l'identique et les mots de deux lettres sont juste inversés (`"TA"` devient `"AT"`). Écrire :

- la fonction `code` qui, étant donné un mot, renvoie le mot recodé : `code("TGV")` renvoie `"GVT"`.
- la fonction `codeDico` qui, étant donné deux noms de fichier, recopie tous les mots du premier dans le second, en les codant.
- le `main` qui fait une copie codée du fichier `dico.txt` vers le fichier `dicoCode.txt`.