Exercices : tableaux à une dimension

Série A: une boucle for

A1. Écrire un algorithme qui affiche un tableau à une dimension d'entiers après l'avoir initialisé de telle manière que chaque élément soit égal à son indice.

A2. Écrire un algorithme qui affiche un tableau à une dimension d'entiers après l'avoir initialisé avec des valeurs lues au clavier.

A3. Écrire un algorithme qui recherche le maximum d'un tableau à une dimension d'entiers initialisé à sa déclaration. L'algorithme affichera le tableau avant le résultat.

A4. Écrire un algorithme qui recherche le maximum et le nombre de fois qu'il apparaît dans un tableau à une dimension d'entiers initialisé à sa déclaration. L'algorithme affichera le tableau avant les résultats.

A5. Écrire un algorithme qui recherche, dans un tableau à une dimension d'entiers initialisé à sa déclaration, le maximum, le nombre de fois qu'il apparaît et la position où il apparaît pour la première fois. L'algorithme affichera le tableau avant les résultats.

A6. Écrire un algorithme qui recherche, dans un tableau à une dimension d'entiers initialisé à sa déclaration, le maximum, le nombre de fois qu'il apparaît et la position où il apparaît pour la dernière fois. L'algorithme affichera le tableau avant les résultats.

A7. Écrire un algorithme qui calcule, dans un tableau à une dimension d'entiers initialisé à sa déclaration, le nombre d'entiers positifs, le nombre d'entiers négatifs et le nombre de zéros de ce tableau. L'algorithme affichera le tableau avant les résultats.

A8. Écrire un algorithme qui calcule, dans un tableau à une dimension d'entiers initialisé à sa déclaration, la somme des éléments du tableau. L'algorithme affichera le tableau avant les résultats.

Série B: deux boucles for juxtaposées

B1. Écrire un algorithme qui calcule la moyenne des valeurs contenues dans un tableau à une dimension de nombres décimaux ainsi que le nombre de ces valeurs supérieures ou égales à la moyenne.

Série C: deux boucles for imbriquées

C1. Soit t un tableau à une dimension d'entiers initialisé à sa déclaration. Écrire un algorithme qui affiche successivement et sur des lignes différentes, le premier élément du tableau, puis les deux premiers, puis les trois premiers et ainsi de suite jusqu'à la totalité du tableau. L'algorithme affichera le tableau avant les résultats.

Série D: shift et rotate (left et right) de une ou plusieurs positions

D1. Écrire un algorithme qui opère un décalage à gauche (shift left) de 1 position des éléments d'un tableau à une dimension d'entiers initialisé à sa déclaration. Le tableau sera affiché avant et après le décalage. Le dernier élément du tableau conserve sa valeur (variante: prend la valeur 0).

D2. Modifier l'algorithme de l'exercice précédent pour qu'il opère un décalage à gauche de n positions (n étant un entier positif). La valeur de n sera lue au clavier et supposée correcte.

D3. Écrire un algorithme qui opère un décalage à droite (shift right) de 1 position des éléments d'un tableau à une dimension d'entiers initialisé à sa déclaration. Le tableau sera affiché avant et après le décalage. Le premier élément du tableau conserve sa valeur (variante: prend la valeur 0).

D4. Modifier l'algorithme de l'exercice précédent pour qu'il opère un décalage à droite de n positions (n étant un entier positif). La valeur de n sera lue au clavier et supposée correcte.

D5. Écrire un algorithme qui opère une rotation à gauche (rotate left) de 1 position des éléments d'un tableau à une dimension d'entiers initialisé à sa déclaration. Le tableau sera affiché avant et après la rotation.

D6. Modifier l'algorithme de l'exercice précédent pour qu'il opère une rotation à gauche de n positions (n entier positif). La valeur de n sera lue au clavier et supposée correcte.

D7. Écrire un algorithme qui opère une rotation à droite (rotate right) de 1 position des éléments d'un tableau à une dimension d'entiers initialisé à sa déclaration. Le tableau sera affiché avant et après la rotation.

D8. Modifier l'algorithme de l'exercice précédent pour qu'il opère une rotation à droite de n positions (n entier positif). La valeur de n sera lue au clavier et supposée correcte.

Série E: recherche linéaire

E1. (**recherche linéaire**)Écrire un algorithme qui recherche dans un tableau à une dimension d'entiers la position d'une valeur entière lue au clavier. Le tableau sera initialisé à sa déclaration et affiché. On supposera que les éléments du tableau ne sont pas triés.

E2. Un tableau à une dimension contient des entiers triés en ordre croissant. Écrire un algorithme qui recherche la position du premier élément du tableau supérieur ou égal à une valeur entière lue au clavier. Si la valeur est supérieure au dernier élément du tableau, l'algorithme fournira comme valeur de la position, l'indice du dernier élément plus 1.

E3. Un tableau à une dimension contient des entiers triés en ordre croissant. Écrire un

algorithme qui recherche la position du dernier élément du tableau inférieur ou égal à une valeur entière lue au clavier. Si la valeur est inférieure au premier élément du tableau, l'algorithme fournira comme valeur de la position, l'indice du premier élément moins 1.

E4. Un tableau à une dimension contient des entiers triés en ordre croissant. Écrire un algorithme qui calcule le nombre d'entiers du tableau strictement compris entre deux valeurs lues au clavier.

E5. (**insertion**) Soit t un tableau à une dimension d'entiers initialisé à sa déclaration. Il contient N éléments mais seuls les N-1 éléments sont initialisés. De plus, les éléments sont rangés en ordre croissant. Soit nbr un entier quelconque lu au clavier. Écrire un algorithme qui insère nbr dans le tableau t de manière à ce que les éléments de celui-ci restent rangés en ordre croissant. L'algorithme affichera le tableau avant de lire la valeur du nombre à insérer. Il affichera le tableau après l'insertion. Exemple : t = [1, 3, 4, 7, 9, 10] et nbr = 5 ⇒ t = [1, 3, 4, 5, 7, 9, 10]

Série F: recherche dichotomique

F1. (**recherche dichotomique**) Comme le E1 mais les éléments du tableau sont triés en ordre croissant.

Série G: gestion simultanée de plusieurs tableaux

G1. Soit t un tableau à une dimension d'entiers initialisé à sa déclaration. Écrire un algorithme qui transfère les entiers positifs de t dans un tableau tp, et les entiers négatifs dans un tableau tn. Les tableaux tp et tn seront affichés. L'algorithme calculera également le nombre de zéros du tableau t. Le tableau t ne sera parcouru qu'une seule fois. L'algorithme commencera par afficher le tableau t.

G2. Soit u un tableau à une dimension d'entiers initialisé à sa déclaration. Écrire un algorithme qui transfère les éléments de u dans un tableau v suivant le principe suivant : les éléments de rang impair de u sont rangés dans v en ordre inverse en commençant par la fin de v, et les éléments de rang pair de u sont rangés dans v dans le même ordre en commençant par le début de v. L'algorithme commencera par afficher le tableau u. Exemple : u = [-1, 2, 7, 1, 3, 1, -2] et v = [2, 1, 1, -2, 3, 7, -1]. Écrire au moins deux versions de l'algorithme, l'une parcourant le tableau deux fois, l'autre ne le parcourant qu'une seule fois.

G3. Soit u un tableau à une dimension d'entiers non nuls. Soit v un tableau à une dimension de même taille que u et dont les éléments sont 0 ou 1. La compression de u par v est le tableau w à une dimension dont les éléments sont dans l'ordre ceux du tableau u pour lesquels l'élément correspondant (de même indice) de v est 1. Exemple : u = [1, 2, 3, 4, 5, 6, 7] v = [0, 1, 1, 0, 1, 0, 1] w = [2, 3, 5, 7, 0, 0, 0] Écrire un algorithme qui réalise une telle compression. Les tableaux u et v seront initialisés à leur déclaration et affichés avant le tableau w.

G4. Soient u et v deux tableaux à une dimension d'entiers et de même taille. Écrire un algorithme qui, à partir des deux tableaux u et v, construit les tableaux tMin et tMax vérifiant la propriété suivante : quelle que soit la valeur de l'indice i, tMin[i] est le

minimum de u[i] et v[i], tMax[i] est le maximum de u[i] et v[i] De plus l'algorithme calculera le nombre d'égalités (u[i] = v[i]). Les deux tableaux u et v seront initialisés à leur déclaration et affichés avant les résultats.

G5. (**fusion**) Écrire un algorithme qui fusionne deux tableaux à une dimension d'entiers triés en ordre croissant dans un troisième tableau à une dimension. Les deux premiers tableaux seront initialisés à leur déclaration, n'auront pas nécessairement la même taille et seront affichés.

G6. Soit un tableau à une dimension d'entiers initialisé à sa déclaration. Écrire un algorithme qui recherche et mémorise les positions où apparaît le maximum de ce tableau. Suggestion: utiliser un deuxième tableau pour mémoriser les différentes positions du maximum. Il est permis d'écrire deux algorithmes, l'un parcourant le tableau deux fois, l'autre ne le parcourant qu'une seule fois.

G7. On donne un tableau à une dimension contenant dans un ordre quelconque les 20 premiers entiers positifs. Écrire un algorithme qui construit un deuxième tableau dans lequel chaque élément a pour valeur la position de son indice dans le premier tableau. A l'aide de ces deux tableaux, l'algorithme affichera la suite des 20 premiers entiers positifs.

Série H: opérations sur des ensembles représentés par des tableaux

H1. (**intersection**) Deux tableaux à une dimension contiennent des entiers triés en ordre croissant. Écrire un algorithme qui sauve dans un troisième tableau les éléments communs aux deux premiers tableaux. A tout moment de l'algorithme, les éléments du troisième tableau restent triés en ordre croissant.

H2. (**union**) Deux tableaux à une dimension contiennent des entiers triés en ordre croissant. Écrire un algorithme qui sauve dans un troisième tableau les éléments qui appartiennent à l'un ou l'autre des deux premiers tableaux. A tout moment de l'algorithme, les éléments du troisième tableau restent triés en ordre croissant.

H3. (**différence symétrique**) Deux tableaux à une dimension contiennent des entiers triés en ordre croissant. Écrire un algorithme qui sauve dans un troisième tableau les éléments qui n'appartiennent qu'à un seul des deux premiers tableaux (autrement dit les éléments qui ne sont pas communs à ces deux tableaux). A tout moment de l'algorithme, les éléments du troisième tableau restent triés en ordre croissant..

Série I: plateaux

I1. Un plateau est une suite de valeurs égales. Une série de plateaux de nombres entiers positifs ou nuls sont introduits au clavier. La fin de la série est marquée par la valeur -1. Écrire un algorithme qui calcule la longueur et la valeur du plus long plateau de cette série.
- Transformer l'algorithme pour qu'il lise les données à partir d'un tableau (initialisé à sa déclaration) et non à partir du clavier
- dans le cas où la fin de la série est encore marquée par la valeur -1
- dans le cas où c'est la fin du tableau qui marque la fin de la série

Série J: mettre de l'ordre dans un tableau (sans le trier)

J1. Un tableau à une dimension contient des entiers égaux à 0, 1 ou 2. Écrire un algorithme qui range les éléments de ce tableau dans un deuxième tableau de manière que les 0 soient rangés en tête, puis les 1 puis les 2. Le premier tableau ne sera parcouru qu'une seule fois. Il sera initialisé à sa déclaration et affiché.

J2. Résoudre le problème précédent sans utiliser de deuxième tableau. Le traitement sera effectué directement dans le premier tableau.

J3. On donne un tableau à une dimension d'entiers et un nombre entier nbr lu au clavier. Écrire un algorithme qui range les éléments du tableau dans un deuxième tableau de manière que les toutes les valeurs inférieures à nbr soient rangées en tête, puis les valeurs égales à nbr, puis les valeurs supérieures à nbr. Le premier tableau ne sera parcouru qu'une seule fois. Il sera initialisé à sa déclaration et affiché.

J4. Résoudre le problème précédent sans utiliser de deuxième tableau. Le traitement sera effectué directement dans le premier tableau.

Série K: divers

K1. Écrire un algorithme qui, à partir d'une année entrée comme un entier de la forme AAAA (donc de 4 chiffres) et du numéro d'ordre du jour dans l'année, fournit la date exprimée sous la forme du numéro d'ordre du jour dans le mois, suivi du nom du mois et de l'année. L'algorithme tiendra compte des années bissextiles. Une année est bissextile si elle est un multiple de 4 sans être un multiple de 100 ou si elle est un multiple de 400. Les longueurs des 12 mois de l'année seront rangées dans un tableau initialisé à sa déclaration. De même les noms des 12 mois de l'année seront rangés dans un tableau initialisé à sa déclaration.

K2. Écrire un algorithme qui, à partir d'une date entrée comme un nombre entier de la forme JJMMAAAA, calcule le numéro d'ordre du jour dans l'année.

K3. Écrire un algorithme qui calcule le nombre de jours séparant deux dates données d'une même année. Les deux dates sont entrées comme des nombres entiers de la forme JJMMAAAA (donc de 8 chiffres).

K4. On donne trois tableaux d'entiers triés en ordre croissant et de tailles différentes. Écrire un algorithme qui recherche la plus petite valeur commune à ces trois tableaux.

K5. Des enfants sont rangés en cercle et chantent une comptine de m syllabes. La première syllabe désigne le premier enfant, la seconde le suivant et ainsi de suite. L'enfant sur lequel tombe la dernière syllabe est éliminé et sort du cercle. La comptine reprend avec l'enfant suivant et se répète jusqu'à ce qu'il ne reste plus qu'un seul enfant. Si le cercle contient au départ n enfants numérotés de 1 à n, quel sera le numéro d'ordre du dernier enfant dans le cercle?

K6. Soit t un tableau à une dimension d'entiers initialisé à sa déclaration. Écrire un

algorithme qui range dans le même tableau les éléments dans l'ordre inverse. L'algorithme commencera par afficher le tableau t et n'utilisera pas de tableau auxiliaire. Exemple : t = [1, 2, 3, 0, 4, -1] ⇒ t = [-1, 4, 0, 3, 2, 1]

K7. Soit un tableau à une dimension d'entiers dont chaque élément est égal à son indice. Écrire un algorithme qui désordonne le tableau de la manière suivante:
   a) choisir de manière aléatoire un nombre entier compris 0 et n (soit i ce nombre),
   b) permuter les éléments d'indices i et n,
   c) décrémenter n d'une unité,
   d) reprendre en a) avec la nouvelle valeur de n.

K8. Écrire un algorithme qui génère de manière aléatoire tous les nombres entiers compris entre 0 et n. L'algorithme ne générera pas deux fois le même nombre. Remarque: il n'est pas interdit de s'inspirer de l'exercice précédent.

Exercices : tableaux à deux dimensions

L1. Écrire un algorithme qui calcule la somme de tous les éléments d'un tableau à deux dimensions d'entiers. Il y aura deux versions de l'algorithme, l'une comportant un parcours ligne par ligne du tableau, l'autre un parcours colonne par colonne.

L2. Soit un tableau à deux dimensions d'entiers. Calculer pour chaque ligne et pour chaque colonne la somme de leurs éléments. Ces sommes seront rangées dans 2 tableaux à 1 dimension, un pour les lignes et un pour les colonnes. Calculer également la somme de tous les éléments du tableau. L'algorithme affichera les 3 tableaux ainsi que la somme totale sous la forme d'une balance carrée.

L3. Soit un tableau à deux dimensions d'entiers. Écrire un algorithme qui remplace les valeurs initiales des éléments d'un sous-tableau de ce tableau par des zéros. Le sous-tableau est déterminé par les indices de ses coins supérieur gauche et inférieur droit. Ces indices seront lus au clavier.

L4. Comme le précédent, mais le rectangle est déterminé par les indices de son coin supérieur gauche, sa longueur et sa hauteur.

L5. Comme les deux précédents, mais seul le périmètre du rectangle sera garni de zéros.

L6. Initialiser un tableau carré d'entiers avec des 1 sur la diagonale principale et des 0 partout ailleurs. Le tableau sera affiché ligne par ligne.

L7. Initialiser un tableau carré d'entiers avec des 1 sur la diagonale secondaire et des 0 partout ailleurs. Le tableau sera affiché ligne par ligne.

L8. Initialiser un tableau carré d'entiers avec des 1 sur les diagonales principale et secondaire et des 0 partout ailleurs. Le tableau sera affiché ligne par ligne.

L9. Écrire un algorithme qui transfert dans un tableau à une dimension d'entiers, les éléments d'un tableau à deux dimensions d'entiers. Le transfert se fera ligne par ligne.

L10. Écrire un algorithme qui inverse les lignes et les colonnes d'un tableau carré. (Cette opération est appelée transposition). Exemple : après transposition, le tableau

1 2 3		**1 4 7**
4 5 6	devient	**2 5 8**
7 8 9		**3 6 9**

L11. Écrire un algorithme qui effectue une rotation de 90° dans le sens trigonométrique sur un tableau carré. Exemple : après rotation le tableau

1 2 3		**3 6 9**
4 5 6	devient	**2 5 8**
7 8 9		**1 4 7**

L12. Écrire un algorithme qui réalise la spécification suivante : sur un échiquier, c'est-à-dire un tableau de 8 lignes et 8 colonnes, à partir d'une case quelconque, marquer

toutes les cases susceptibles d'être atteintes en un coup par une tour. Au terme de l'exécution, les cases accessibles en un coup contiendront la valeur 1, les autres la valeur 0. La position de la tour est lue au clavier.

L13. Même exercice que le précédent mais pour un fou.

L14. Soit un damier rectangulaire. Un pion part du coin supérieur gauche et ne peut se déplacer que vers la droite ou vers le bas. Par combien de chemins différents peut-il rallier le coin inférieur droit? Remarque: nombre de chemins permettant d'accéder à une case donnée = nombre de chemins conduisant à la case située immédiatement à gauche + nombre de chemins conduisant à la case immédiatement supérieure.