

Devoir de synthèse N°3		
Lycée El Amal Fouchena Lycée Mghira Lycée M'hamédia 01 Lycée M'hamédia 02	Section : Sciences de l'informatique	
	Matière : Algorithmique et programmation	
Niveau : 4 ^{ème} année	Date : Jeudi 11 Mai 2023	Durée : 3 Heures
Nom & Prénom :	Classe :	Note : /20

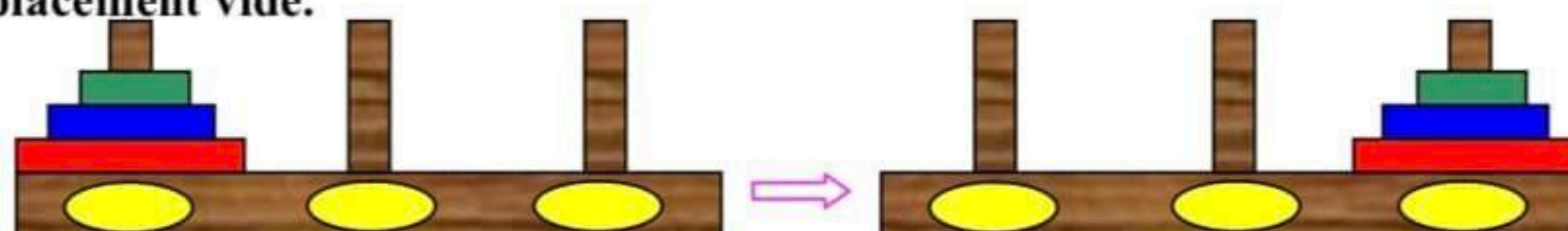
NB : Cette page est à remplir et à remettre avec le sujet.

Tout module développé doit être accompagné des déclarations nécessaires.

Exercice 1 (4 points)

Le problème des **tours de Hanoï** est un jeu de réflexion consistant à déplacer des disques de diamètres différents d'une tour de « **départ** » à une tour d'« **arrivée** » en passant par une tour « **intermédiaire** » et ceci en un minimum de coups tout en respectant les règles suivantes :

- On ne peut déplacer plus d'un disque à la fois
- On ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.



Soit la procédure Hanoi suivante qui permet de simuler le jeu décrit précédemment :

Procédure Hanoi (nb_disques : entier, tour_origine, tour_destination, tour_intermediaire: caractère)

Début

Si nb_disques > 0 alors

Hanoi (nb_disques - 1, tour_origine, tour_intermediaire, tour_destination)

Écrire ('déplacer ', tour_origine, ' → ', tour_destination)

Hanoi (nb_disques - 1, tour_intermediaire, tour_destination, tour_origine)

Fin si

Fin

Travail demandé : Pour chacune des propositions suivantes, cocher la bonne réponse :

1. Qu'est-ce que la récursivité?

- ☐ Une technique de programmation qui consiste à appeler un module à l'intérieur de lui même
- ☐ Une technique de programmation qui consiste à appeler un module à l'extérieur du même module
- ☐ Une technique de programmation qui consiste à appeler un module à l'intérieur d'un autre module

2. Dans une fonction récursive, que se passe-t-il si la condition d'arrêt n'est pas atteinte ?

- ☐ La fonction continue à s'appeler elle-même indéfiniment
- ☐ La fonction retourne une erreur
- ☐ La fonction se termine sans exécuter les instructions restantes

3. Dans la fonction Hanoi, quelle est la condition d'arrêt :

- ☐ nb_disques > 0
- ☐ nb_disques = 0
- ☐ nb_disques < 0

4. Quels sont les déplacements à réaliser si on appelle la procédure Hanoi (2, "A", "B", "C")

- ☐

```
entrer le nombre de disques 2  
  
déplacer de A vers C  
déplacer de A vers B  
déplacer de C vers B
```
- ☐

```
entrer le nombre de disques 2  
  
déplacer de A vers B  
déplacer de A vers C  
déplacer de B vers C
```
- ☐

```
entrer le nombre de disques 2  
  
déplacer de C vers B  
déplacer de C vers A  
déplacer de B vers A
```

5. Le cas d'exécution suivant est relatif à quel appel de la procédure Hanoi :

```
entrer le nombre de disques 3  
  
déplacer de A vers B  
déplacer de A vers C  
déplacer de B vers C  
déplacer de A vers B  
déplacer de C vers A  
déplacer de C vers B  
déplacer de A vers B
```

- ☐ Hanoi (3, "A", "C", "B")
- ☐ Hanoi (3, "A", "B", "C")
- ☐ Hanoi (3, "C", "B", "A")
6. Quel est le nombre minimal pour déplacer N disques :
- ☐ 2^n
- ☐ $2^n - 1$
- ☐ $2^n + 1$

7. On observe que pour un nombre de disques un peu grand, le nombre de déplacements à effectuer augmente. La solution est de sauvegarder les déplacements ainsi que leur nombre dans un fichier texte. Ecrire un module qui permet de modifier la procédure Hanoi afin de sauvegarder les déplacements nécessaires dans un fichier texte.

NB : Le nombre de déplacements doit être enregistré à la dernière ligne du fichier texte stocké physiquement sous c:\bac2023\dep.txt), précédée par les différents déplacements effectués.

L'élève n'est pas appelé à ouvrir le fichier, il est déjà ouvert et fermé dans le programme appelant.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Exercice 2 (3 points)

En additionnant un nombre N avec le nombre composé des mêmes chiffres de N mais disposés dans un sens inverse et en recommençant le même traitement, on obtient dans une majorité des cas un nombre palindrome.

Ecrire un algorithme d'un module qui permet de vérifier si on peut obtenir un nombre palindrome à partir d'un entier N , tout en appliquant la méthode décrite ci-dessus et avec un nombre d'itérations inférieur ou égal à k .

NB :

- La solution doit comporter au moins un module récursif.
- N et k sont saisis dans le programme appelant.

Exemples pour un $K=5$

- 124 nécessite une seule itération pour obtenir un nombre palindrome : $124 + 421 = 545$
- 59 nécessite 3 itérations pour obtenir un nombre palindrome :
1^{ère} itération : $59 + 95 = 154$
2^{ème} itération : $154 + 451 = 605$
3^{ème} itération : $605 + 506 = 1111$
- Pour $N=89$ la génération d'un nombre palindrome n'est pas possible au bout de 5 itérations
1^{ère} itération : $89 + 98 = 187$
2^{ème} itération : $187 + 781 = 968$
3^{ème} itération : $968 + 869 = 1837$
4^{ème} itération : $1837 + 7381 = 9218$
5^{ème} itération : $9218 + 8129 = 17347$
→ Génération non accomplie après 5 itérations

Exercice 3: (4 points)

- 1- Ecrire un algorithme d'une fonction intitulée **Premier_Suivant** qui prend en paramètre un entier a premier et retourne l'entier premier qui lui suit.

Exemple :

Premier_Suivant (23) retourne 29 (car 29 est l'entier premier qui suit l'entier premier 23).

- 2- Soit S une constante qui est égale à :

$$S = \frac{3}{1!} - \frac{5}{2!} + \frac{7}{3!} - \frac{11}{4!} + \frac{13}{5!} - \frac{17}{6!} + \frac{19}{7!} - \frac{23}{8!} \dots$$

En utilisant la fonction **Premier_Suivant**, écrire un algorithme d'un module qui permet de déterminer une valeur approchée de S à ε près.

Exercice 4 : (4 points)

Le filtre de médian fonctionne en remplaçant la valeur de chaque pixel dans l'image par la médian des pixels dans un voisinage donné. La taille de ce voisinage est déterminée par la taille du filtre. Par exemple, un filtre de médian de taille 3x3 utilise un voisinage de 9 pixels (3x3), tandis qu'un filtre de moyenne de taille 5x5 utilise un voisinage de 25 pixels (5x5).

L'application d'un filtre de médian peut réduire le bruit dans une image, mais elle peut également entraîner une perte de détails fins. Il est donc important de choisir judicieusement la taille du filtre en fonction.

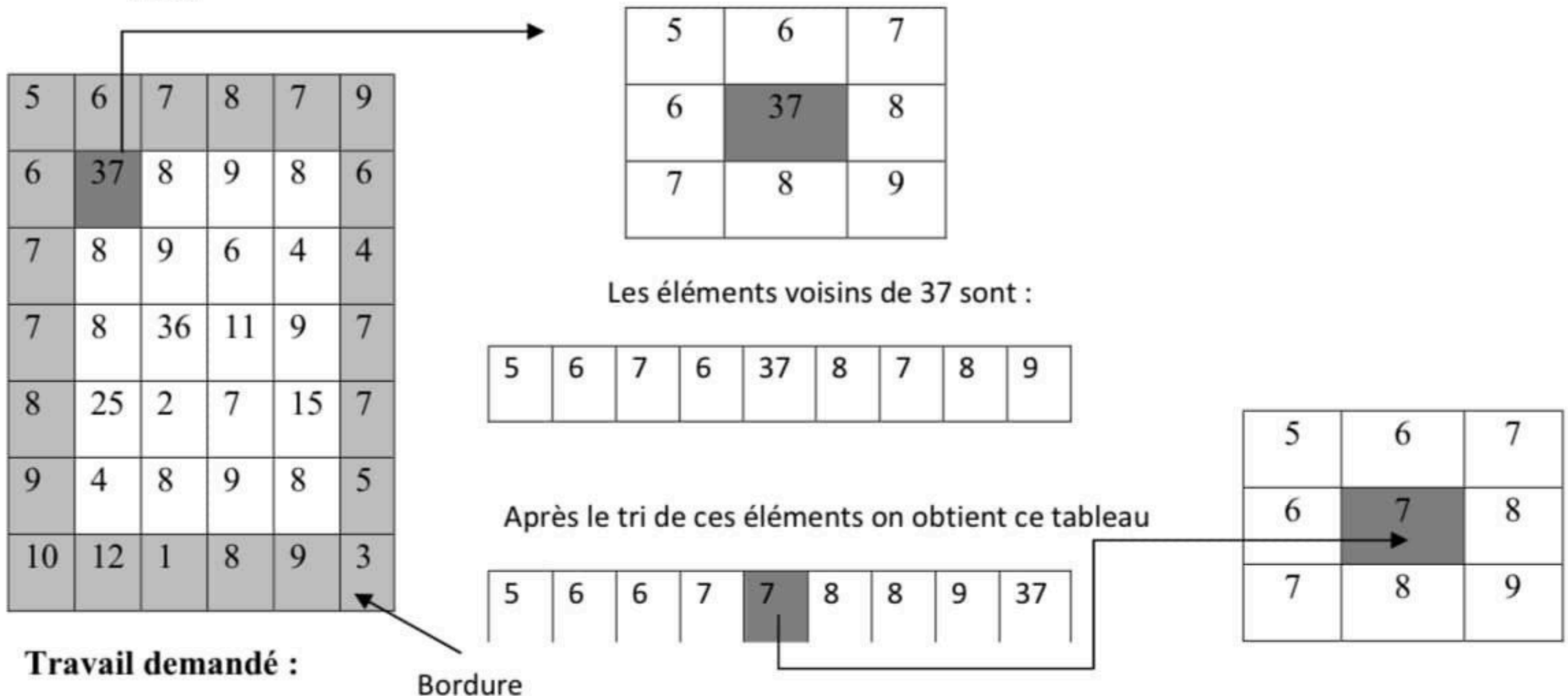


Principe :

L'idée principale du filtre médian est de remplacer chaque élément par la valeur médiane de son voisinage.

Par exemple, pour un filtre de taille 3 et si on considère ces neuf pixels, dont une valeur forme un bruit dans cette image (ici 37), elle sera remplacée par la valeur médiane (ici 7).

MI :



Travail demandé :

Ecrire l'algorithme d'un module qui permet d'appliquer l'effet médian de taille 3 à une image au niveau de gris (tableau à deux dimensions **MI** de taille **L** et **C** et de type **MAT**) et de générer une nouvelle matrice **MF** en appliquant le principe décrit ci-dessus.

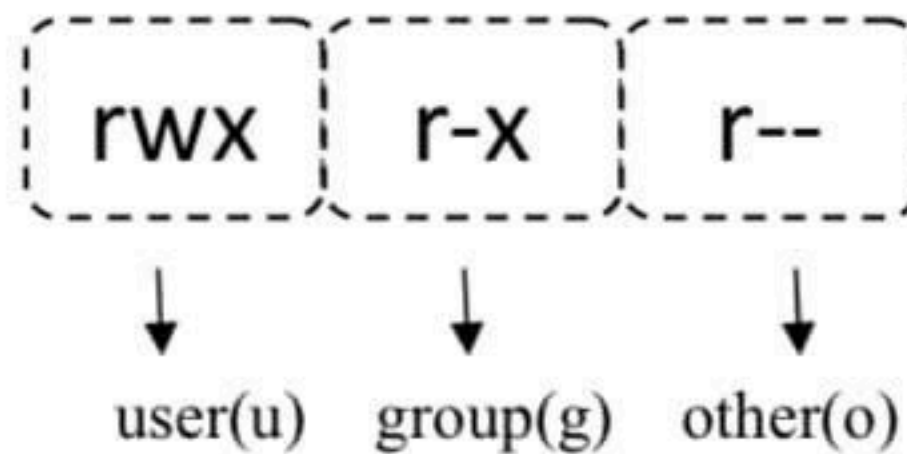
N.B : Les éléments de la bordure ne doivent subir aucune modification.

*L'élève n'est pas appelé à remplir la matrice initiale **MI** représentant l'image*

Exercice 5 : (5 points)

Le système d'exploitation Linux permet de définir des droits d'accès différents sur un même fichier selon la catégorie de l'utilisateur. A chaque catégorie d'utilisateurs : **user(u)**, **group(g)** ou **other(o)**, on associe un triplet de droits : lecture (**r**), écriture (**w**) et exécution (**x**).

En effet, dans l'**exemple** suivant, la représentation des droits d'accès à un fichier est illustrée comme suit :



L'**exemple** précédent se traduit comme suit :

- Si la catégorie de l'utilisateur est **user**, il disposera de tous les droits (r, w et x).
- Si la catégorie de l'utilisateur est **group**, il disposera uniquement des droits de lecture (r) et d'exécution (x). Le droit d'écriture est refusé, il est donc représenté par le caractère "-".
- Si la catégorie de l'utilisateur est **other**, il disposera uniquement du droit de lecture (r). Les autres droits ont été refusés.

D'autre part, la commande **chmod** permet de changer les droits d'accès à un fichier. Elle présente la forme générale suivante : `chmod expression nom_fichier` où **expression** contient une liste des utilisateurs (u, g, o) suivie par la liste des droits (r, w, x) à ajouter ou à retirer. Un droit est ajouté à l'aide du signe "+" et est retiré à l'aide du signe "-".

Exemple :

Etant donnés les droits d'accès associés à un fichier intitulé **document.doc** suivants : **rw-r--r-x**

- La commande **chmod ug-x document.doc** permettra de retirer des utilisateurs (**user**) et (**group**) le droit d'exécution sur le fichier **document.doc**

Ainsi, les droits d'accès après la modification seront : **rw-r--r-x**

- La commande **chmod o+w document.doc** permettra de donner aux utilisateurs (**other**) le droit d'écriture sur le fichier **document.doc**.

Ainsi, les droits d'accès après la modification seront : **rw-r--rwx**

- La commande **chmod ugo+rw document.doc** permettra de donner tous les droits à tous les utilisateurs.

Ainsi, les droits d'accès après la modification seront : **rw-rwxrwx**

Travail demandé :

Etant donnés :

- un fichier de données intitulé **droits.dat** dont chaque bloc contient un enregistrement formé par 4 champs :
 - ✓ trois chaînes de trois caractères chacune représentant les catégories d'utilisateurs **user(u)**, **group(g)** ou **other(o)**.
 - ✓ et une chaîne représentant le nom d'un fichier.
- et un fichier texte intitulé **chmod.txt**, dont chaque ligne contient la commande **chmod** suivi d'un espace suivi de l'expression suivi d'un espace suivi du nom du fichier.

On se propose d'écrire un module permettant d'appliquer les commandes se trouvant dans le fichier **chmod.txt** sur les droits d'accès enregistrés dans le fichier **droits.dat** et stocker les nouveaux droits dans un nouveau fichier de données de même structure que le fichier **droits.dat** intitulé **resultat.dat**.

NB : Les fichiers **droits.dat**, **chmod.txt** et **resultat.dat** se trouvent sur la racine du disque C:

*L'élève n'est pas appelé à remplir les fichiers **droits.dat**, **chmod.txt**.*

Exemple :

Pour le contenu du fichier **droits.dat** suivant :

```
rwXr--rw- _ reseau.txt
r---w-r-x _ tableau.xls
--X---r-- _ cours.doc
rwXr-xrwx _ exercice.docx
```

et le contenu du fichier **chmod.txt** suivant :

```
Chmod _ ugo-wx _ reseau.txt
Chmod _ go+wx _ cours.doc
Chmod _ uo+rwX _ tableau.xls
```

Le contenu du fichier **resultat.dat** après application de la commande **chmod** sera le suivant :

```
r--r--r-- _ reseau.txt
rwx-w-rwx _ tableau.xls
--X-wXrwx _ cours.doc
rwXr-xrwx _ exercice.docx
```