

REPUBLIQUE TUNISIENNE MINISTERE DE L'EDUCATION ◆◆◆ EXAMEN DU BACCALAUREAT SESSION DE JUIN 2014	Epreuve : ALGORITHMIQUE ET PROGRAMMATION
	Durée : 3 H
	Coefficient : 2,25
Section : Sciences de l'informatique	Session de contrôle

Exercice 1 : (2 points)

Soit le tableau de déclaration suivant :

Objet	Type/Nature
F1	Fichier d'entiers
F2	Texte
x	Entier

Dans le tableau ci-dessous, valider chacune des instructions suivantes, en complétant la deuxième colonne par **Valide** ou **Invalide** tout en justifiant la réponse, dans la troisième colonne, dans le cas où l'instruction est invalide.

Instructions	Valide / Invalide	Justifications
Lire (F1)		
Ecrire_nl (F2,x)		
Pointer (F1,x)		
$x \leftarrow \text{Taille_fichier} (F2)$		

Exercice 2 : (3,5 points)

Une chaîne de caractères est dite palindrome si et seulement si on peut la lire de la même façon de droite à gauche comme de gauche à droite. Exemples : RADAR, KAYAK.

Travail demandé :

Ecrire un algorithme d'un module permettant de vérifier si une chaîne de caractères **CH**, est transformable en une chaîne palindrome ou non. Dans l'affirmative, afficher une transformation de CH en une chaîne palindrome sinon afficher le message "*transformation impossible*".

N. B : On suppose que la chaîne CH est déjà saisie et formée uniquement de lettres majuscules.

Exemples :

- pour la chaîne "ZIAZA", le module affichera la chaîne "AZIZA" ou la chaîne "ZAI AZ".
- pour la chaîne "TOUTE", le module affichera "transformation impossible".

Exercice 3 : (3,5 points)

Afin de calculer le coût de la réalisation d'un lac artificiel, un paysagiste a besoin de déterminer sa surface, qui est représentée par l'aire S délimitée par les courbes de deux fonctions f et g qui se croisent en deux points A et B , ayant respectivement les abscisses -1 et $2/3$, comme le montre la figure (Fig. 1) ci-dessous.

Les deux fonctions f et g sont définies par :

- $f(x) = x^2 + 2x + 1$
- $g(x) = -5x^2 + 5$

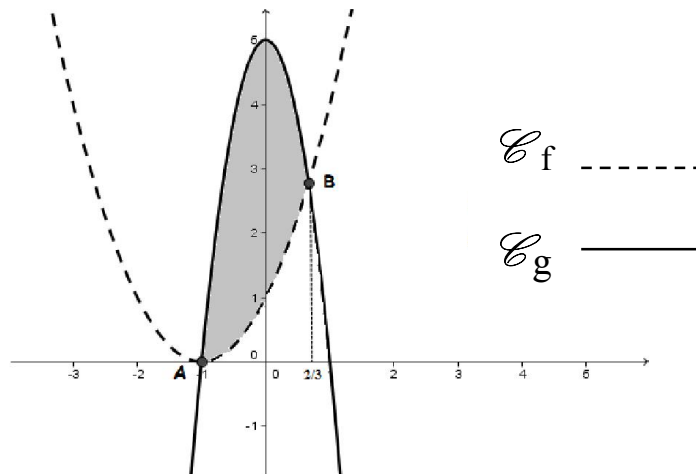


Fig. 1

Travail demandé :

Ecrire un algorithme d'un module permettant de déterminer une valeur approchée de l'aire S délimitée par les deux courbes des deux fonctions f et g définies dans l'intervalle $[A, B]$.

Problème : (11 points)

Une image représentée en niveaux de gris, est formée par des pixels codés chacun sur huit (8) bits et ayant chacun une valeur comprise entre 0 (noir) et 255 (blanc).

Le filtre médian est une technique de filtrage utilisée dans les traitements d'images bitmap.

Exemple :



Image à filtrer

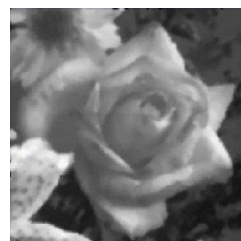


Image filtrée

L'algorithme du filtre médian consiste à remplacer chaque valeur du pixel de l'image à filtrer par la valeur médiane des pixels voisins. Il s'agit d'un voisinage carré autour du pixel considéré.

Un voisinage est donc assimilable à une matrice carrée de taille impaire, exemple : voisinage 3×3 , 5×5 , 7×7 , etc. (voir l'exemple ci-après)

N. B. : Lorsqu'un pixel est proche du bord de l'image, certains pixels de son voisinage n'existent pas.

En disposant d'un fichier "**ImageInit.txt**" contenant les codes binaires correspondants aux pixels d'une image à filtrer, on se propose d'écrire un programme permettant de remplir une matrice **MF**, par les valeurs, en décimal, des pixels de l'image filtrée par le principe du filtre médian (voisinage 3x3).

Sachant que :

1. Le fichier "**ImageInit.txt**" est constitué comme suit :

- la première ligne, contient la définition de l'image sous la forme **HxL** où **H** est le nombre de pixels en hauteur et **L** le nombre de pixels en largeur.

Exemple : "200x180"

- chacune des **HxL** lignes suivantes, contient une succession de huit (8) bits représentant le code binaire d'un pixel de l'image.

2. L'image filtrée sera obtenue en procédant comme suit :

- remplir à partir du fichier "**ImageInit.txt**", une matrice **M** de **H** lignes et **L** colonnes par les conversions en décimal des blocs de huit (8) bits représentant chacun le code d'un pixel.
- appliquer un filtre médian sur chaque élément **M[i,j]** de la matrice **M** afin de former une deuxième matrice **MF** et ce de la manière suivante :

- placer les valeurs de la matrice **M** formant le voisinage 3x3 de **M[i,j]**, dans un tableau **T** (y compris le pixel lui-même).

N. B. : les éléments du voisinage qui sont en dehors de la matrice **M**, auront la valeur zéro dans le tableau **T**.

- trier par ordre croissant les éléments du tableau **T**.

- placer la valeur située au milieu du tableau **T** trié (valeur médiane), dans la case **MF[i,j]** de la matrice **MF**.

Exemple :

M	1	2	3	4	5	6
1	7	10	12	7	9	9
2	12	8	7	11	9	10
3	9	9	10	8	7	12
4	10	9	9	6	10	9
5	12	12	12	10	7	7
.....

Pour le pixel M[2,1] :

- Placement des valeurs du voisinage 3x3 y compris le pixel à traiter

T	1	2	3	4	5	6	7	8	9
	0	7	10	0	12	8	0	9	9

- Tri du tableau T pour déterminer la valeur médiane

T	1	2	3	4	5	6	7	8	9
	0	0	0	7	8	9	9	10	12

- Placement de la valeur médiane dans **MF[2,1]**

MF	1	2	3	4	5	6
1
2	8
3	9
4
5
.....

Pour le pixel M[3,5] :

- Placement des valeurs du voisinage 3x3 y compris le pixel à traiter

T	1	2	3	4	5	6	7	8	9
	11	9	10	8	7	12	6	10	9

- Tri du tableau T pour déterminer la valeur médiane

T	1	2	3	4	5	6	7	8	9
	6	7	8	9	9	10	10	11	12

- Placement de la valeur médiane dans **MF[3,5]**

Travail demandé :

- 1) Analyser ce problème en le décomposant en modules.
- 2) Analyser chacun des modules envisagés.