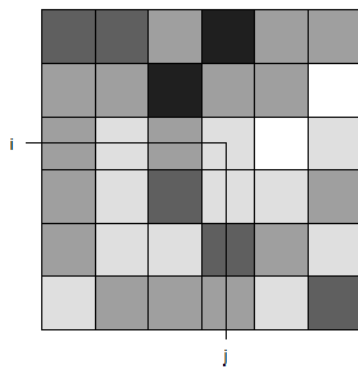


Devoir surveillé <input type="checkbox"/>	Examen <input checked="" type="checkbox"/>	Session : principale <input checked="" type="checkbox"/> de contrôle <input type="checkbox"/>
Matière : Algorithmique et structures de données	Semestre: 1	
Enseignant : Wided Miled et Aymen Sellaouti	Date: 03/01/2018 à 15h00	
Filière(s) : MPI	Durée: 1h30	
Nombre de pages : 3	Documents : autorisés <input type="checkbox"/> non autorisés <input checked="" type="checkbox"/>	

### Exercice 1 (8 pts)

Une image en niveaux de gris est un tableau à deux dimensions. Chaque case de ce tableau représente un pixel de l'image. La valeur de ce pixel est le niveau de gris associé.



89	89	120	20	120	120
120	120	20	120	120	255
120	181	120	181	255	181
120	181	89	181	181	120
120	181	181	89	120	181
181	120	120	120	181	89

Image en niveaux de gris

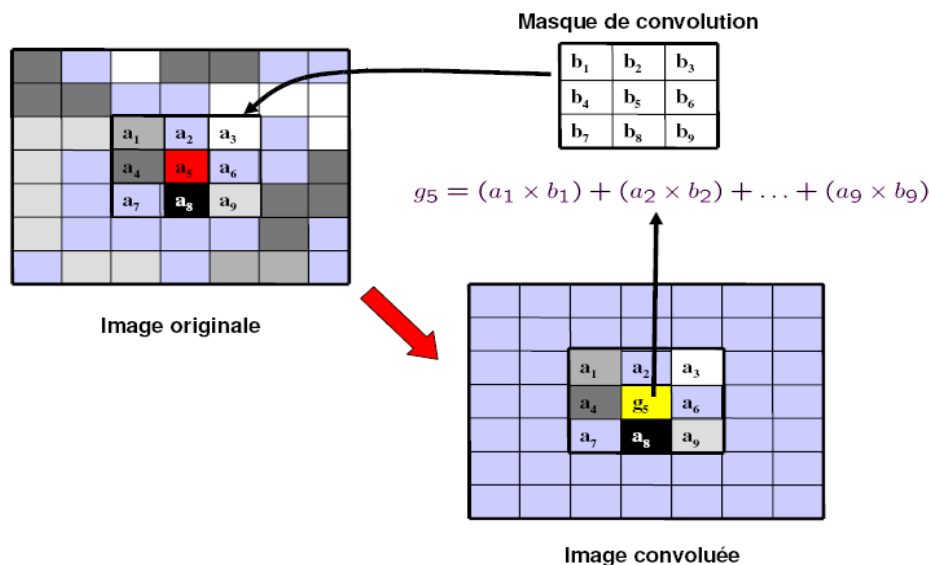
niveaux de gris

Nous voulons modéliser en algorithmique quelques fonctionnalités basiques sur les images.

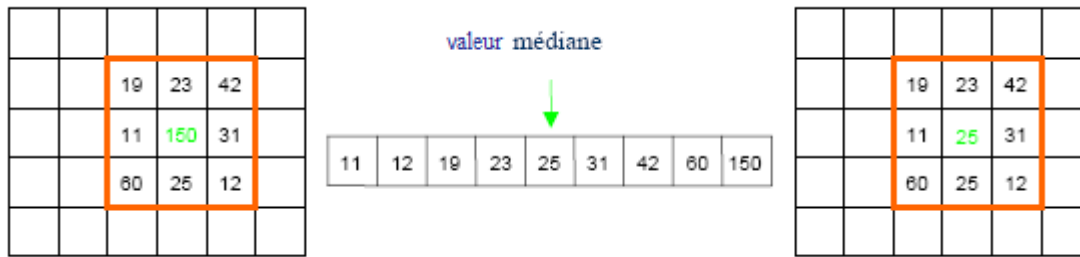
1. Ecrire en algorithmique la fonction ou procédure **ConvolPoint( )** qui permet de convoluer un **point de l'image** par un filtre représenté par son masque de convolution et de retourner la valeur calculée.

La fonction ConvolPoint( ) prendra comme paramètres le tableau 2D représentant l'image, le masque de convolution et les coordonnées i et j du point à convoluer. On suppose ici qu'un filtre (le masque de convolution) est un tableau à deux dimensions de taille 3\*3.

Convoler un point de l'image par ce filtre revient à faire l'opération décrite par l'image ci-dessous.



2. Nous voudrions maintenant créer une fonction qui nous permet d'appliquer un filtre médian à l'image. Le principe du filtre Médian consiste à remplacer la valeur d'un pixel de coordonnées i et j par la valeur médiane de la répartition des niveaux de gris des pixels voisins. Nous traitons dans cet exemple le point central de niveau de gris 150.



Le principe donc est d'extraire les voisins directs, de les trier puis de retourner la valeur médiane. Ecrire en algorithmique la fonction **MadianPoint()** qui permet à partir d'un tableau 2D, et des coordonnées i et j d'un point de retourner la valeur médiane qui lui correspond.

### Exercice 2 (12 pts) :

Afin de gérer les matchs des phases de poules (premier tour) de la coupe du monde, une startup dispose des fichiers suivants :

#### « **equipe.txt** »

qui contient l'ensemble des équipes ayant participé à la compétition

Chaque ligne de ce fichier contient les données suivantes :

Code equipe	: 5 caractères (entier)
Pays	: 10 caractères
Nomination	: 30 caractères
Groupe	: 1 caractère

#### « **match.txt** »

Qui contient tous les matchs de tous les groupes.

Chaque ligne de ce fichier contient les données suivantes :

Code match	: 5 caractères (entier)
Date	: 8 caractères au format suivant aaaammjj(exemple 20180702)
Code équipe1	: 5 caractères (entier)
Code équipe2	: 5 caractères (entier)

#### « **resultats.txt** »

Qui contient tous les résultats des matchs de tous les groupes.

Chaque ligne de ce fichier contient les données suivantes :

Code match	: 5 caractères (entier)
Résultat	: 1 caractères ( <b>1</b> si la première équipe a gagné, <b>x</b> si pas de vainqueur et <b>2</b> si c'est la deuxième équipe qui a gagné)
Nombre but Equipe1	: 2 caractères (entier)
Nombre but Equipe2	: 2 caractères (entier)

Chaque groupe contient 4 équipes. Dans chaque groupe toutes les équipes s'affrontent ce qui fait que chaque équipe joue trois matches.

En cas de victoire, une équipe récupère 3 points, en cas de match nul (même nombre de but pour les 2 équipes), l'équipe récupère 1 point et en cas de défaite l'équipe récupère 0 point.

La différence de but d'une équipe est la différence de buts entre les buts qu'elle a marqué et les buts qu'elle a encaissé durant tous ces matches.

Les deux premières équipes de chaque groupe se qualifient pour le tour suivant.

Le rang d'une équipe se base sur le nombre de points. Plus une équipe a de points mieux elle est classée.

Si deux équipes ont le même nombre de points alors l'équipe qui a la meilleure différence de buts est mieux classée.

Si deux équipes ont le même nombre de points et la même différence de but, on les classera par ordre alphabétique de Nomination.

**NB : On suppose qu'on ne peut jamais avoir trois équipes ayant le même nombre de point.**

Afin d'aider la startup, on vous demande d'écrire les fonctions suivantes :

1- **void get\_equipes\_groupes**(FILE \*fp, char groupe, int\* t){ }

qui retourne dans le tableau t les 4 codes des équipes appartenant au groupe passé en paramètre.

2- **void total\_butts\_differenceDeButs** (FILE \*fp, int code\_equipe, int\* points, int\* differenceButs) { }

qui permet de retourner le nombre de point d'une équipe ainsi que sa différence de buts.

3- Ecrire un **programme C** qui à permet à l'utilisateur de saisir un groupe et lui affiche le pays et la nomination des deux équipes qualifiées. En cas de besoin, ajouter les fonctions et structures qui vous semblent nécessaires pour avoir un code clair et bien organisé.

4- Ecrire la fonction qui permet de mettre à jour le résultat d'un match en changeant le résultat et le nombre de buts des deux équipes.

**void actualise\_resultat\_match**(FILE \*fp, int code\_match, char nouv\_result, int nouv\_nb\_but\_equipe1, int nouv\_nb\_but\_equipe2) { }

**Bonne chance ☺**