

Contrôle Continu Terminal : Mercredi 20 Janvier 2010

Les documents, calculatrice, téléphone portable, mini-ordinateur, pda, ... sont interdits
La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.
Vous veillerez à respecter les notations et les règles d'écriture des algorithmes vues en cours et en TD. Soyez attentifs aux indices des tableaux et prenez garde à ne pas dépasser la fin des tableaux.
Le barème est donné à titre indicatif.

Partie A : QCM (5 pts)

La feuille jointe en annexe devra être complétée et rendue avec votre copie.
N'oubliez pas d'y inscrire le **numéro d'anonymat** fourni sur la copie.

Partie B : Algorithmique (7 pts)

1. On appelle nombre de Armstrong un entier naturel qui est égal à la somme des cubes des chiffres qui le composent.
Exemple: $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27$, est un nombre de Armstrong.
Écrire l'algorithme d'un sous-programme permettant d'afficher tous les nombres d'Armstrong compris entre 1 et 10000.
2. Écrire l'algorithme d'un sous-programme permettant de supprimer toutes les occurrences d'un caractère passé en paramètre dans une chaîne de caractères.
Exemple : `enleve_car ("cestbientotlesvacances", 'e') → cstbintotlsvacancs`

Partie C : Langage C (8 pts)

On souhaite développer un logiciel permettant d'effectuer des traitements simples sur les images : extraction de valeurs caractéristiques, seuillage, miroir horizontal, addition.



image initiale

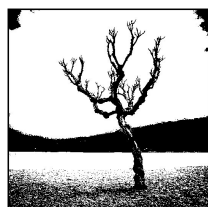


image seuillée



miroir image



initiale + miroir

Une image est constituée d'une grille (tableau) de pixels représentant une intensité lumineuse (niveau de gris). Le niveau de gris d'un pixel est une valeur entière comprise entre 0 (noir) et 255 (blanc).

Afin de pouvoir gérer des images de tailles variables, on définira une taille maximale MAX_X et MAX_Y pour les tableaux utilisés.

1. Écrire en langage C / C++ la structure de données permettant de stocker une image. Une image est caractérisée par 3 informations : taille-x (nombre de pixels sur une ligne), taille_y (nombre de pixels sur une colonne) et un tableau contenant les intensités lumineuses de chacun des pixels.
2. Écrire en langage C / C++ un sous-programme permettant d'extraire, en un seul parcours de l'image :
 - le niveau de gris minimum de l'image,
 - le niveau de gris maximum de l'image,
 - le niveau de gris moyen de l'image
3. L'opération dite de "seuillage simple" consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur (appelée seuil) et à la valeur maximale les pixels ayant une valeur supérieure. Ainsi le résultat du seuillage est une image binaire ne contenant que des pixels noirs et blancs.
Écrire en langage C / C++ un sous-programme permettant d'effectuer le seuillage d'une image. Le seuil choisi par l'utilisateur sera passé en paramètre.
4. Écrire en langage C / C++ un sous-programme permettant d'effectuer la symétrie verticale d'une image (miroir). Le résultat sera stocké dans une nouvelle image qui sera retournée au programme principal.

Exemple sur un tableau 3*3 :

1	2	3
4	5	6
7	8	9

3	2	1
6	5	4
9	8	7

5. Écrire en langage C / C++ un sous-programme permettant de retourner une nouvelle image, calculée comme étant la somme de deux images passées en paramètres. Chaque nouveau pixel est calculé à partir des pixels correspondants des images sources. Lorsque la somme des intensités lumineuses des deux pixels ajoutés est supérieure à 255, on la fixera à cette valeur limite.

Question Bonus (2 points)

6. Écrire en langage C / C++ un sous-programme permettant de proposer le menu suivant à l'utilisateur (et d'appeler les sous-programmes correspondants)

== MENU ==
1- Quitter
2- Afficher intensités min, max et moyenne
3- Seuiller l'image
4- Symétrie de l'image
5- Somme de deux images