

Question (4 points)

Écrire un sous-programme qui, à partir d'un tableau d'entiers, crée un tableau de nombres pairs et un tableau de nombres impairs.

Par exemple, du tableau [2, 4, 3, 5, 9, 10, 6, 7, 8], on obtient le tableau de nombres pairs [2, 4, 10, 6, 8] et celui de nombres impairs [3, 5, 9, 7].

Problème (16 points)

Le capteur de vision d'un robot domestique génère une ligne de 256 pixels de couleur. Ces pixels sont ensuite interprétés pour créer la vision artificielle du robot. Chaque pixel est représenté par sa couleur et son intensité lumineuse. Les couleurs sont représentées en format rouge, vert, bleu. Chaque composante (rouge, verte ou bleu) est un nombre entier entre 0 et 255. L'intensité lumineuse du pixel est une valeur réelle dans l'intervalle [0, 100].

Structure Couleur		
Champ	Description	Type
rouge	Composante rouge d'une couleur (0 à 255)	Entier
vert	Composante verte d'une couleur (0 à 255)	Entier
bleu	Composante bleue d'une couleur (0 à 255)	Entier

Structure Pixel		
Champ	Description	Type
couleur	Couleur d'un pixel	Couleur
intensité	Intensité lumineuse (0 à 100) d'un pixel	Entier

1. En vous aidant des tableaux décrivant les types *Couleur* et *Pixel* ci-dessus, écrivez les structures *Couleur* et *Pixel*.

2. Ecrire, en langage algorithmique :

- la fonction *moyenneIntensité(V capteur() Pixel, V tailleCapteur Entier) Réel* permettant de calculer l'intensité lumineuse moyenne du capteur.
- la fonction *estNoir(V p Pixel) Logique* qui indique si un pixel est noir ou non. Un pixel est noir si les trois composantes de sa couleur sont à 0 ou si son intensité lumineuse est nulle.
- la fonction *consécutifs(V capteur() Pixel, V tailleCapteur Entier) Entier* qui retourne le plus grand nombre de pixels "noirs" consécutifs sur une ligne du capteur.
- la procédure *tri(R capteur() Pixel, V tailleCapteur Entier)* qui trie la ligne du capteur en fonction des intensités lumineuses croissantes.