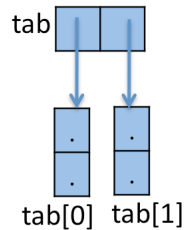


TP Langage C

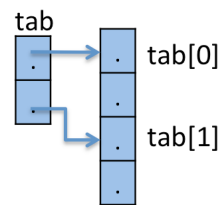
L'objectif de ce TP est de se familiariser avec les tableaux et les pointeurs au travers de la manipulation de matrices.

1 Exercice 1 : création

- Écrivez un programme C qui lit sur l'entrée standard les paramètres réels d'une matrice carrée de dimension 2 et les stocke dans un tableau $tab[2][2]$ alloué statiquement. Le programme affichera sur la sortie standard les éléments de la matrice sous la forme : $tab[i][j] = valeur$.
- Même exercice avec cette fois ci un tableau à 2 dimensions alloué dynamiquement : `**tab`.



(1)



(2)

1. Dans un premier temps, on allouera dynamiquement les colonnes (ou les lignes) de manière indépendantes (voir la figure (1) ci-dessus).
2. Dans un deuxième temps, on allouera dynamiquement un vecteur de taille suffisante pour stocker l'ensemble des éléments de la matrice et on mettra à jour le vecteur `tab[]` (voir la figure (2) ci-dessus). Commentez sur la différence entre ce tableau et le tableau alloué statiquement dans l'exercice précédent.

Le tableau sera libéré en fin de programme à l'aide de la fonction `free()`.

2 Exercice 2 : passage de paramètres

- Ajoutez au programme précédent une fonction ... *determinant(...)* qui prend en paramètre une matrice carrée de dimension 2 et retourne la valeur de son déterminant (le programme affichera cette valeur) :
 1. la matrice est représentée par un tableau alloué statiquement.
 2. la matrice est représentée par un tableau alloué dynamiquement.
- Ajoutez au programme une fonction ... *transposee(...)* qui prend en paramètre une matrice carrée de dimension 2 et retourne la matrice transposée $tab_transposée[i][j] = tab[j][i]$ (le programme affichera cette matrice) :
 1. la matrice transposée est créée au sein de la fonction (allocation dynamique).
 2. la matrice transposée est créée au sein du programme (allocation dynamique) et passée en deuxième paramètre de la fonction.
 3. Laquelle des deux solutions vous semble la plus pertinente pour gérer de manière cohérente la libération de la mémoire allouée dynamiquement ?

3 Exercice 2 : tableaux de structures

On souhaite remplacer dans le programme précédent les réels par un type structuré.

- Déclarez le type structuré ETUDIANT qui contient deux membres : une chaîne de caractères *nom* et une chaîne de caractère *filière*.
- Modifiez le programme précédent de manière à lire et stocker un tableau de 4 étudiants alloués dynamiquement. Le programme affichera par ailleurs ce tableau.
- Comment pourrait-on réécrire la fonction *transposee* de l'exercice précédent pour qu'elle fonctionne avec un tableau d'étudiants ?