

## TD 6 - Allocation dynamique et tableaux dynamiques

**Exercice 1** Ecrire une fonction `lireTableau` qui étant donné un entier  $n$  retourne un tableau de  $n$  entiers saisis au clavier.

**Exercice 2** On souhaite utiliser un tableau dynamique pour représenter le tableau bi-dimensionnel statique suivant : `int ts[7][5]`;

Quatre instructions d'allocation sont proposées :

1. Alice propose l'instruction : `int *td = (int *)malloc(7*5*sizeof(int));`
2. Bernard propose l'instruction : `int *td = (int *)malloc(sizeof(int[7][5]));`
3. Camille propose :  
`int **td = (int **)malloc(7*sizeof(int*));`  
`for(int i=0;i<7;i++) td[i]=(int *)malloc(5*sizeof(int));`
4. Dominique propose l'instruction : `int (*td)[5] = (int(*)[5])malloc(sizeof(int[7][5]));`

Dites pour chacune des propositions :

1. Combien d'octets sont alloués ? On fera un schéma représentant la mémoire allouée.
2. Comment accède-t-on à la case de la 3<sup>ème</sup> ligne et 2<sup>ème</sup> colonne ?
3. Quelles instructions permettent de libérer la mémoire occupée par le tableau ?

Laquelle parmi ces propositions se rapproche le plus de la vision dynamique du tableau statique `ts` ?

**Exercice 3** Écrire une fonction `extrairePairs` qui étant donné un tableau d'entiers `td` et sa taille, retourne un tableau `tr` ne contenant que les entiers pairs de `td`. Bien réfléchir à la signature de cette fonction qui doit permettre ultérieurement d'exploiter correctement le tableau retourné.

**Exercice 4** Écrire alors une fonction `main` qui :

1. saisit un nombre  $n$  au clavier ;
2. utilise la fonction `lireTableau` pour créer un tableau `t1` de  $n$  entiers ;
3. utilise la fonction `extrairePairs` pour récupérer les entiers pairs dans un tableau ;
4. "supprime" le tableau `t1` ;
5. affiche le contenu du tableau des pairs.
6. libère la mémoire et termine.

Que faut-il modifier dans ce programme, si on remplace l'étape 2 de création de `t1` par la ligne `int t1[TAILLE]={1,3,4,14,11,8,10,4}`; (en ajoutant en début de fichier `#define TAILLE 8`) ?

**Exercice 5** On souhaite se doter d'un type vecteur qui possède toutes les caractéristiques des tableaux dynamiques mais ayant en plus la capacité de changer de taille. Écrire une fonction `changeTaille` qui prend trois paramètres : un tableau dynamique, sa taille, et la nouvelle taille désirée et retourne vrai si la modification a été possible (le tableau a donc été retaillé) ou faux si cela a échoué (le tableau n'a pas été modifié). La *stdlib* propose une fonction *realloc* qui faciliterait l'implantation de cette fonction mais qui n'est pas à utiliser ici !