Feuille de TD n°8 de Programmation 2

(Algorithmique élémentaire des matrices de flottants)

Dans cette feuille, on manipule des matrices de flottants en utilisant la structure présentée en cours le 2 mai.

```
struct mat_float {
  int lig; /* nombre de lignes de la matrice */
  int col; /* nombre de lignes de la matrice */
  double ** tab_coeff; /* tableau dynamique bidimensionnel des coefficients */
};
```

Dans la suite, « matrice » abrégera « structure `mat_float` ». On été vues en cours les fonctions suivantes.

```
/** Alloue sur le tas l'espace pour une matrice nulle à l lignes et c colonnes, */
struct mat_float *creer_initialiser_matrice (int l, int c);

/** Supprime la ligne d'indice l et la colonne d'indice c de la matrice *m */
void reduire_matrice (struct mat_float *m, int l, int c);
```

Exercice 1. Addition et multiplication de matrices

a) Écrivez la définition d'une fonction `additionner_matrice` qui reçoit en entrées les adresses de deux matrices $m1$ et $m2$, et renvoie l'adresse d'une nouvelle matrice égale à la somme $m1 + m2$ quand la somme est possible. Évaluez en fonction des dimensions de $m1$ et $m2$ le nombre d'opérations scalaires requises par son exécution.

b) Écrivez la définition d'une fonction `multiplier_matrice` qui reçoit en entrées les adresses de deux matrices $m1$ et $m2$, et renvoie l'adresse d'une nouvelle matrice égale au produit $m1 \times m2$ quand le produit est possible. Évaluez en fonction des dimensions de $m1$ et $m2$ le nombre d'opérations scalaires requises par son exécution.

Exercice 2. Calcul du déterminant par la formule de Laplace

Soit $M = (m_{i,j})_{0 \leq i,j \leq n-1}$ une matrice carrée de dimension n et soit j un indice de colonne. Le déterminant de M peut être calculé par la formule de Laplace :

$$\det M = \sum_{i=0}^{n-1} (-1)^{i+j} m_{i,j} \det M_{i,j},$$

$M_{i,j}$ désignant la matrice obtenue en supprimant la ligne d'indice i et la colonne d'indice j de M. Les $\det M_{i,j}$ sont des *mineurs* de M.

a) Écrivez la définition d'une fonction **récursive** `determinant_matrice` qui reçoit en entrée l'adresses d'une matrice m et renvoie la valeur de son déterminant si la matrice est carrée. Des appels à `reduire_matrice` faciliteront le calcul des mineurs de M. Éstimez en fonction de la dimension de M le nombre d'opérations scalaires requises requises par l'exécution de cette fonction.

b) Montrez qu'on peut se passer des appels à `reduire_matrice` dans la fonction de la question précédente et ainsi « atténuer » son inefficacité en supprimant presque tous les appels aux fonctions d'allocation de mémoire sur le tas.