Partiel IF122 - Durée : 50mn

Les documents sont interdits. Vous n'avez pas le droit de sortir de la salle avant la fin de l'épreuve. Votre nom doit figurer en commentaire tout au début de votre programme, sous la forme : // NOM Prénom. Une fois le temps ecoulé, vous quitterez la salle sans éteindre votre machine le temps que l'un des surveillants vienne récupérer votre résultat. Bonne chance.

Une matrice se représente habituellement à l'aide d'un tableau. Ceci a deux inconvénients. Le premier est que la taille du tableau doit être fixée à priori, et le second, que chacune des cases ai une valeur. Or bien souvent les grandes matrices ont beaucoup de cases vides (de valeur zéro). Nous allons les représenter à l'aide d'une liste chaînée, avec pour convention de ne pas faire apparaître les zéro.

Cette liste doit correspondre à la lecture des éléments de la matrice de gauche à droite en partant du coin supérieur gauche, puis on continue en revenant à la ligne. Les cellules contiennent à la fois la valeur de la case `val`, mais aussi ses deux coordonnées `x,y` respectivement pour la colonne et la ligne.

Les types vous sont donnés :

```
#include <iostream>
typedef struct cellule{
  int val;
  int x;
  int y;
  cellule * suiv;
};
typedef cellule* liste_M;
```

Question 1 : Pour tester vos fonctions, vous aurez besoin d'un exemple. Écrivez une fonction `exemple` qui retourne une liste correspondant à la matrice :

$$\begin{pmatrix} 5 & -7 \\ 0 & 4 \end{pmatrix}$$

Question 2 : On veut afficher cette liste. Pour la matrice de l'exemple précédent, le résultat doit être :
$(1, 1, 5)$ $(2, 1, -7)$
$(2, 2, 4)$

C'est à dire que l'on affiche, par lignes, les cellules.
Écrivez une procédure qui affiche une `liste_M`. Essayez la sur notre exemple.

Les questions suivantes visent à calculer, sur notre représentation, la transposée de la matrice. C'est à dire que l'on va échanger les lignes et les colonnes.

Question 3 : Écrire une fonction qui retourne le numéro de la plus petite colonne qui contient un élément non nul. Elle retourne `-1` si la matrice est nulle. Vérifiez la sur notre exemple.

Question 4 : On suppose que l'on dispose d'une procédure `extrait_transpose_colonne(liste_M &m,int i,liste_M &a,liste_M &b)` qui extrait de `m` la colonne numéro `i`, en plaçant le résultat transposé dans `a`. `b` est un pointeur vers le dernier élément de `a`. Utilisez cette procédure et la fonction précédente pour écrire une procédure récursive qui effectue la transposée d'une matrice.

Question 5 : Écrivez à présent la procédure `extrait_transpose_colonne(liste_M &m,int i,liste_M &a,liste_M &b)`. Faites attention à placer les élément de `a` dans le bon ordre, et d'échanger leur coordonnées.

Question 6 : Vérifier sur notre exemple.