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.

Le type 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.

Question 3 : Si on cherche à additionner deux matrices, un cas très simple est celui où l'on ajoute une matrice qui est réduite à une seule cellule. On suppose que l'on a à notre disposition une procédure `ajoute_reduite(liste_M m,cellule &c)`, qui peut lors de l'ajout modifier les champs de `c`.

Écrivez une procédure récursive simple `additionne_M` qui utiliserait `ajoute_reduite` pour additionner deux listes représentant des matrices.

Question 4 : Il faut maintenant écrire `ajoute_reduite(liste_M m,cellule &c)`. Il faut faire attention à la chose suivante : si m contient une cellule de mêmes coordonnées que `c` et de valeur opposée, alors le résultat de la somme est 0, or par convention une telle cellule ne doit pas apparaître dans la représentation de la matrice.

Question 5 : Testez votre procédure `ajoute_reduite` en ajoutant la cellule $(2, 1, 7)$ à notre exemple. Testez l'addition de matrice en additionnant notre exemple avec lui même.