Travaux Dirigés de C n°7

Cours de langage et programmation

—L3 Informatique—

On souhaite écrire un programme permettant de manipuler des nombres de diverses natures. Pour cela, on utilise le type `number` déclaré comme suit :

```
#ifndef NUMBERS_H
#define NUMBERS_H

typedef enum{INT_NUMBER,COMPLEX_NUMBER,ERROR} number_type;

typedef struct NB number;

typedef union{
    int intval;
    double doublePair[2];
  } value_type;

struct NB{
  number_type type;
  value_type val;
  number (*add)();
  number (*mult)();
  number (*getUnit)(void);
  number (*getZero)(void);
};

number add(const number,const number);
number mult(const number,const number);
number getUnit(const number);
number getZero(const number);

#endif
```

▸ **Exercice 1.** *Écrire le fichier qui contient la définition d'une fonction* `number getIntNumber(const int);` *qui construit un* `number` *à partir d'un entier.*

▸ **Exercice 2.**
Écrire la définition des fonctions `add` *et* `getZero` *déclarées dans le fichier numbers.h*

▸ **Exercice 3.**
Définir de façon similaire aux entiers une implémentation de `number` *pour les nombres complexes. Enrichir le type* `number` *de manière à disposer d'une fonction* `display` *pour chaque* `number`.

▸ **Exercice 4.**

- *Écrire une fonction* `allocMatrix` *prenant en argument un* `number` x *et un* `int` n *et retournant une matrice carrée de taille* n *dont les coefficients sont des* `number` *de même type que* x *(on supposera la matrice initialisée à 0). Éviter les informations redondantes de sorte à ne pas gaspiller de mémoire.*
- *Écrire des fonctions getCoeffMatrix et setCoeffMatrix permettant d'accéder en lecture et en écriture aux coefficients d'une matrice.*
- *Écrire une fonction* `freeMatrix` *prenant en argument une matrice carrée et libérant l'espace qui lui est alloué.*
- *Écrire une fonction* `howManyMatrices` *donnant le nombre de matrices allouées à cet instant. Écrire ceci de manière à ne manipuler aucune variable visible en dehors du module* `Matrix`*, puis, le cas échéant, modifier le code de sorte à n'avoir aucune variable globale.*
- *Créer un nouveau type de nombre :* `Matrix` *et écrire les fonctions nécessaires.*

▸ **Exercice 5.**

Écrire une fonction `intPower` *qui calcule la puissance (entière positive) d'un* `number`*. Afficher la puissance* 5*-ème de*

$$\begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$$