

TRAVAUX DIRIGÉS - 3

EXERCICE 1

Dans cet exercice, nous allons implémenter et utiliser quelques outils d'algèbre linéaire.

**Q. 1** Définir le type `vector` correspondant à une structure ayant pour champs :

- `dim` : dimension du vecteur (type `int`),
- `pval` : pointeur sur les composantes du vecteur (type `double *`). ■

**Q. 2** Nous voulons écrire une fonction `AllocVector` permettant d'allouer un objet de type `vector` à la dimension `N`.

1. Donner deux prototypes possibles (non trivialement similaire) pour la fonction `AllocVector`.
2. Choisir un prototype et écrire la fonction associée. ■

**Q. 3** 1. Écrire une fonction `freeVector` permettant de désallouer le champ `val` d'un objet de type `vector`.

2. Justifier le choix de ses arguments! ■

**Q. 4** 1. Écrire une fonction `FREEVector` permettant de désallouer le champ `val` d'un objet de type `vector`, de mettre à zéro le champ `dim` et à `NULL` le champ `val`.

2. Justifier le choix de ses arguments! ■

**Q. 5** Nous voulons écrire une fonction `RandVector` permettant d'initialiser chacune des composantes d'un vecteur par un nombre aléatoire (utiliser la fonction `drand48()` du langage C qui retourne un `double` entre 0 et 1 : `double drand48 (void);`). Le vecteur est supposé déjà alloué.

1. Donner deux prototypes possibles (non trivialement similaire) pour la fonction `RandVector`.
2. Choisir un prototype et écrire la fonction associée. ■

**Q. 6** Soient  $v \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^n$ ,  $\alpha \in \mathbb{R}$  et  $\beta \in \mathbb{R}$ . Les fonctions `ScalarProd`, `aUpbV` et `Norm` correspondent à :

$$\begin{aligned} \text{ScalarProd} & : (u, v) = \sum_{i=1}^n u_i v_i, \\ \text{aUpbV} & : \alpha u + \beta v, \\ \text{Norm} & : \|u\| = \left( \sum_{i=1}^n u_i^2 \right)^{1/2}. \end{aligned}$$

1. Pour chaque fonction, donner deux prototypes possibles (non trivialement similaire).
2. Pour chaque fonction, `ScalarProd` et `aUpbV`, choisir un prototype et écrire la fonction associée. ■

**Q. 7** Écrire un `main` simple et fonctionnel utilisant au moins les fonctions `RandVector`, `ScalarProd` et `aUpbV`. ■

## EXERCICE 2

Nous avons les structures suivantes :

```
1 typedef struct{
2     int dim;
3     double *val;
4 } vector;
5
6 typedef struct{
7     int nL; /* Nombre de lignes */
8     int nC; /* Nombre de colonnes */
9     double **val;
10 } matrix;
```

Soit  $\mathbb{A}$  une matrice réelle de taille  $N \times M$  et `matrix A`; l'objet qui lui est associé en langage C. On souhaite avoir

$$\begin{aligned} \mathbb{A}.nL &= N \\ \mathbb{A}.nC &= M \\ \mathbb{A}.val[i-1][j-1] &= \mathbb{A}_{i,j}, \forall (i,j) \in \{1, \dots, N\} \times \{1, \dots, M\}. \end{aligned}$$

**Q. 1** Nous voulons écrire une fonction `AllocMatrix` permettant d'allouer/initialiser un objet de type `matrix` à la taille  $N \times M$ .

1. Donner l'ensemble des prototypes de fonction permettant de réaliser cette opération.
2. Quels sont les prototypes de fonction invalides?
3. Choisir un prototype et écrire la fonction associée. ■

**Q. 2** Écrire une fonction `freeMatrix` permettant de desallouer le champ `val` d'un objet de type `matrix`. Justifier de ses arguments! ■

**Q. 3** Nous voulons écrire une fonction `FREEMatrix` permettant de desallouer le champ `val` d'un objet de type `matrix`, de mettre à zéro les champs `nL` et `nV`, et à `NULL` le champ `val`.

1. Donner l'ensemble des prototypes de fonction permettant de réaliser cette opération.
2. Quels sont les prototypes de fonction invalides?
3. Choisir un prototype et écrire la fonction associée. ■

**Q. 4** Nous voulons écrire une fonction `RandMatrix` permettant d'initialiser chacune des composantes par un nombre aléatoire (utiliser la fonction `drand48()` du langage C : `double drand48 (void);`). La matrice est supposée déjà allouée.

1. Donner l'ensemble des prototypes de fonction permettant de réaliser cette opération.
2. Quels sont les prototypes de fonction invalides?
3. Choisir un prototype et écrire la fonction associée. ■

**Q. 5** Soit  $\mathbb{A} \in \mathcal{M}_{n,p}(\mathbb{R})$ ,  $\mathbb{B} \in \mathcal{M}_{p,m}(\mathbb{R})$  et  $\mathbb{D} \in \mathcal{M}_{n,p}(\mathbb{R})$ .

1. Écrire l'ensemble des prototypes des fonctions valides permettant de calculer :  
 $\text{MatrixProd} : \mathbb{Q} = \mathbb{A}\mathbb{B}$ ,  
 $\text{MatrixSum} : \alpha\mathbb{A} + \beta\mathbb{D}$
2. Écrire l'ensemble des prototypes invalides.
3. Pour chaque fonction, choisir un prototype valide et écrire la fonction associée. ■

**Q. 6** Écrire un *ch'ti main* utilisant les fonctions `MatrixProd` et `MatrixSum`. ■