



Table des Matières

1	Objectif	1	3	Fonction principale	2
2	Le TA matrix	1			

1 Objectif

Chargement d'une matrice depuis un fichier au format texte ou binaire; sauvegarde d'une matrice dans un fichier.

2 Le TA matrix

Créez un fichier `matrix.h` contenant la déclaration du type matrice

Déclaration du TA et de ses fonctions.

```
1 typedef enum{ TXT, BIN } file_mode_t; // format des fichiers
2
3 typedef struct {
4     double * values;
5     int nbLig, nbCol;
6 } matrix_t;
7
8 matrix_t * consMatrix( int nlig, int ncol );
9 matrix_t * cpyMatrix(matrix_t * M );
10 void freeMatrix(matrix_t ** ptrM);
11
12 matrix_t * scanMatrix();
13 void printMatrix( matrix_t * M, char * entete);
14
15 matrix_t * readMatrix(char * filename, file_mode_t mode);
16 void writeMatrix(char * filename, file_mode_t mode, matrix_t * M);
17
18 matrix_t * matrixAdd( matrix_t * A, matrix_t * B);
19 matrix_t * matrixMult( matrix_t * A, matrix_t * B);
```

Servez vous des fonctions de la librairie standard input/output

Les opérateurs des fichiers (librairie `stdio.h`)

```
1 // Ouverture, fermeture du fichier et test de fin de fichier
2 FILE *fopen(const char *pathname, const char *mode);
3 int fclose(FILE *stream);
4 int feof(FILE *stream);
5
6 // lecture, écriture depuis/dans un fichier en mode texte ou binaire
7 int fprintf(FILE *stream, const char *format, ...);
8 int fscanf(FILE *stream, const char *format, ...);
9 int fread(void * ptr, int size_element, int nb_element, FILE * stream);
10 int fwrite(void * ptr, int size_element, int nb_element, FILE * stream);
11
12 // d'autres fonctions qui ne sont pas utiles pour ce TP
13 int fseek(FILE *stream, long offset, int whence);
14 long ftell(FILE *stream);
15 void rewind(FILE *stream);
16 int fgetpos(FILE *stream, fpos_t *pos);
17 int fsetpos(FILE *stream, const fpos_t *pos);
```

Pour connaître leurs fonctionnements, utilisez la commande `man`;

Le fichier **texte** d'une matrice aura la forme suivante

```
3 4
1.1 1.2 1.3 1.4
2.1 2.2 2.3 2.4
3.1 3.2 3.3 3.4
```

où la première ligne est la dimension de la matrice et les lignes suivantes les valeurs rangées... ligne par ligne

Créer le fichier `matrix.c` pour y ranger les définitions des fonctions déclarées dans `matrix.h`.

3 Fonction principale

Créez un fichier `main.c` qui contiendra uniquement la fonction `main(int argc, char * argv[])` qui devra :

- a) Appeler la fonction `readMatrix(argv[1], TXT)` pour créer la matrice pointée par A
- b) Appeler la fonction `readMatrix(argv[2], TXT)` pour créer la matrice pointée par A
- c) Visualiser ces deux matrices,
- d) Si l'addition est possible,
 - Calculez la,
 - Visualisez le résultat,
 - Sauvegardez le résultat au format binaire
 - Supprimez le résultat
 - Charger le résultat depuis le fichier binaire et visualisez le à nouveau
- e) Si la multiplication est possible, faites de même.