

Exercice 1. Dans cet exercice, les fichiers de description de matrice sont organisés de la manière suivante :

- La première ligne est un entier indiquant le nombre de lignes de la matrice.
- La seconde ligne est un entier indiquant le nombre de colonnes de la matrice.
- Les lignes suivantes comportent autant d’entiers que le nombre de colonnes.
- Il y a en tout 2 lignes de plus que le nombre de lignes de la matrice.

Par exemple, le fichier `mat1.txt` contient :

```
2
3
50 12 10
-3 41 15
```

Il désigne la matrice

$$\begin{pmatrix} 50 & 12 & 10 \\ -3 & 41 & 15 \end{pmatrix}$$

Remarque. Connaissant le nombre de lignes et de colonnes on aurait pu se contenter de lister les coefficients sans passage à la ligne.

1. On souhaite vérifier de visu si un fichier est bien une description de matrice. Écrire une fonction `void cat(char * filename)` qui affiche à l’écran le contenu d’un fichier texte.

2. Écrire une procédure `int ** create_matrix(char* nom, int * nbl, int * nbc)` qui prend en paramètres un nom de fichier de description de matrice et deux pointeurs d’entiers.

Cette fonction met à jour les pointeurs d’entiers selon les informations lues dans le fichier : le premier pointe sur le nombre de lignes, le second sur le nombre de colonnes.

La fonction crée ensuite une matrice, met à jour ses coefficients selon les informations lues dans le fichier puis la retourne.

3. Écrire une procédure `void transpose(char * nom, int ** tab, int nbl, int nbc)` qui prend en paramètres un nom de fichier de sortie, un pointeur de pointeur d’entiers désignant une matrice et les dimensions d’icelle. La procédure écrit dans le fichier de sortie la transposée de la matrice passée en paramètre.

Exercice 2. 1. Écrire une fonction `void traiter (int argc, char** argv)` qui prend en paramètres un entier `argc` et un tableau de chaînes de caractères (autant que la valeur de `argv`).

Si le tableau contient plus de deux chaînes, le programme s’arrête sur une erreur d’assertion. Sinon, c’est que le tableau contient exactement 2 chaînes. Ce sont deux noms de fichiers. La fonction copie le contenu du premier dans le second.

2. Modifier le `main` de votre projet de sorte que le programme obtenu affiche son propre nom puis lise les noms des deux fichiers en ligne de commande.

```
$ cat toto1.txt
Toto lé motivé
Gogo lé fatigué
```

```
La Dodo, salé bon mém!  
$ ./copier toto1.txt toto_copy.txt  
Nb d'arguments : 3, fichier source : toto1.txt, cible toto_copy.txt  
$ cat toto_copy.txt  
Toto lé motivé  
Gogo lé fatigué  
La Dodo, salé bon mém!
```