

## 5 – Calcul matriciel

### Objectifs

- Développement d'algorithmes, structuration et mise au point de programmes complets.
- Manipulation de tableaux bidimensionnels.

### 1.1 Description

On considère une matrice carrée de dimension  $M \times M$  dont les éléments sont rangés dans un tableau `MAT` contenant des *valeurs entières 32-bit signées*.

L'objectif est de réaliser trois fonctions permettant de lire/écrire la valeur d'un élément à une position  $(i, j)$  donnée, et de calculer la transposée de la matrice. Il faudra également réaliser un programme principal qui permette de tester les fonctions.

Dans un premier temps, nous travaillerons avec une matrice de dimension  $3 \times 3$  pour la mise au point :

		1	2	3			1	4	7	
MAT	=	4	5	6		<sup>t</sup> MAT	=	2	5	8
		7	8	9				3	6	9

**Remarque :** En pratique, les valeurs de la matrice sont placées en mémoires à des adresses consécutives.

<code>@MAT</code>	01 00 00 00
<code>@MAT + 4</code>	02 00 00 00
<code>@MAT + 8</code>	03 00 00 00
<code>@MAT + C</code>	04 00 00 00
<code>@MAT + 10</code>	05 00 00 00
<code>@MAT + 14</code>	06 00 00 00
<code>@MAT + 18</code>	07 00 00 00
<code>@MAT + 1C</code>	08 00 00 00
<code>@MAT + 20</code>	09 00 00 00

L'accès à un élément `MAT(i, j)` de la matrice se fera en calculant l'adresse par rapport aux indices  $i$  et  $j$  de la façon suivante : `@MAT + ((j * M + i) * 4)`. Par exemple :

`MAT(2, 0) = 3` qui est à l'adresse `@MAT + ((0 * M + 2) * 4) = @MAT + 8`

`MAT(1, 2) = 8` qui est à l'adresse `@MAT + ((2 * M + 1) * 4) = @MAT + 1C`

Les deux premières fonctions que nous allons écrire correspondent à deux sous-programmes `get` et `set`, qui permettront de lire ou d'écrire un élément de la matrice à la position  $(i, j)$ . Elles seront appelées avec les paramètres suivants : adresse de la matrice, dimension de la matrice ( $M$ ), position horizontale ( $i$ ), position verticale ( $j$ ) et valeur à lire/écrire. La troisième fonction `transpose` prend en paramètres l'adresse de la matrice `MAT`, l'adresse de la matrice transposée `MAT_T`, la dimension de la matrice et calcule la transposée.

## 1.2 Lecture/écriture d'une valeur dans une matrice

Ecrire les fonctions `get` et `set` de lecture et d'affectation de valeur à la position  $(i, j)$ . Tester ces deux fonctions sur quelques exemples. Les paramètres des deux fonctions `get` et `set` sont passés par la pile. Le résultat de la fonction `get` est renvoyé dans `R0`.

## 1.3 Transposée

L'algorithme de transposée consiste à parcourir chaque élément de la matrice et à effectuer une permutation de valeur entre  $MAT(i, j)$  et  $MAT(j, i)$ . Cette permutation consiste à écrire l'élément à l'adresse  $@MAT + ((j * M + i) * 4)$  du tableau `MAT` à l'adresse  $@MAT\_T + ((i * M + j) * 4)$  du tableau `MAT_T`.

Ecrire une fonction `transpose` qui permet de transposer la matrice en utilisant les fonctions `get/set`. Les paramètres de la fonction `transpose` sont passés par la pile.