

LES MATRICES

Utilisation et manipulation

1- Exemple de programme en C qui permet de remplir une matrice et de l'afficher :

```
#include <stdio.h>
#define MAX_ROWS 100
#define MAX_COLS 100

void remplirMatrice(int matrice[MAX_ROWS][MAX_COLS], int lignes, int colonnes) {
    for (int i = 0; i < lignes; i++) {
        for (int j = 0; j < colonnes; j++) {
            printf("Entrez la valeur pour l'élément [%d][%d] : ", i, j);
            scanf("%d", &matrice[i][j]);
        }
    }
}

void afficherMatrice(int matrice[MAX_ROWS][MAX_COLS], int lignes, int colonnes) {
    printf("\nMatrice :\n");
    for (int i = 0; i < lignes; i++) {
        for (int j = 0; j < colonnes; j++) {
            printf("%d ", matrice[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int matrice[MAX_ROWS][MAX_COLS];
    int lignes, colonnes;

    printf("Entrez le nombre de lignes : ");
    scanf("%d", &lignes);
    printf("Entrez le nombre de colonnes : ");
    scanf("%d", &colonnes);

    if (lignes > MAX_ROWS || colonnes > MAX_COLS) {
        printf("Erreur : Dimensions dépassent la taille maximale de la matrice.\n");
        return 1;
    }

    remplirMatrice(matrice, lignes, colonnes);
    afficherMatrice(matrice, lignes, colonnes);

    return 0;
}
```

2- Exemple en C pour effectuer l'addition, la multiplication et la division de deux matrices :

```
#include <stdio.h>
#define ROWS 3
#define COLS 3

void additionMatrices(int mat1[ROWS][COLS], int mat2[ROWS][COLS], int result[ROWS][COLS]) {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            result[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
}
```

```

    }

    void multiplicationMatrices(int mat1[ROWS][COLS], int mat2[ROWS][COLS], int result[ROWS][COLS]) {
        for (int i = 0; i < ROWS; i++) {
            for (int j = 0; j < COLS; j++) {
                result[i][j] = 0;
                for (int k = 0; k < COLS; k++) {
                    result[i][j] += mat1[i][k] * mat2[k][j];
                }
            }
        }
    }

    void divisionMatrices(int mat1[ROWS][COLS], int mat2[ROWS][COLS], float result[ROWS][COLS]) {
        for (int i = 0; i < ROWS; i++) {
            for (int j = 0; j < COLS; j++) {
                if (mat2[i][j] != 0) {
                    result[i][j] = (float)mat1[i][j] / mat2[i][j];
                } else {
                    result[i][j] = 0; // Gestion de la division par zéro
                    printf("Erreur : Division par zéro à [%d][%d]\n", i, j);
                }
            }
        }
    }

    void afficherMatriceInt(int mat[ROWS][COLS]) {
        for (int i = 0; i < ROWS; i++) {
            for (int j = 0; j < COLS; j++) {
                printf("%d ", mat[i][j]);
            }
            printf("\n");
        }
    }

    void afficherMatriceFloat(float mat[ROWS][COLS]) {
        for (int i = 0; i < ROWS; i++) {
            for (int j = 0; j < COLS; j++) {
                printf("%.2f ", mat[i][j]);
            }
            printf("\n");
        }
    }

    int main() {
        int mat1[ROWS][COLS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
        int mat2[ROWS][COLS] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};

        int additionResult[ROWS][COLS];
        int multiplicationResult[ROWS][COLS];
        float divisionResult[ROWS][COLS];

        printf("Matrice 1 :\n");
        afficherMatriceInt(mat1);

        printf("\nMatrice 2 :\n");
        afficherMatriceInt(mat2);

        // Addition
        additionMatrices(mat1, mat2, additionResult);
        printf("\nAddition des matrices :\n");

```

```
    afficherMatriceInt(additionResult);

    // Multiplication
    multiplicationMatrices(mat1, mat2, multiplicationResult);
    printf("\nMultiplication des matrices :\n");
    afficherMatriceInt(multiplicationResult);

    // Division
    divisionMatrices(mat1, mat2, divisionResult);
    printf("\nDivision élément par élément des matrices :\n");
    afficherMatriceFloat(divisionResult);

    return 0;
}
```

1. Recherche d'un élément spécifique dans une matrice :

```
#include <stdio.h>

void searchMatrix(int matrix[3][3], int rows, int cols, int target) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (matrix[i][j] == target) {
                printf("Élément trouvé à la position (%d, %d)\n", i, j);
                return;
            }
        }
    }
    printf("Élément non trouvé\n");
}

int main() {
    int matrix[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
    int target = 5;

    searchMatrix(matrix, 3, 3, target);
    return 0;
}
```