

Exercices en salle 9

Tableaux à deux dimensions

1 Exercices rapides d'appropriation

1. Comment déclarer un tableau à deux dimensions avec 4 lignes et 8 colonnes ?
2. Comment créer une zone de mémoire dynamique pouvant stocker des 7 pointeurs `int*` ?
3. Si on a la variable `char** text`, que représente `text[2][6]` ?
4. Quel est la notation équivalente à `data[7][5]` en utilisant l'arithmétique des pointeurs ?

2 Fonction `createMatrix`

Écrivez une fonction `createMatrix` qui va créer une matrice avec `M` lignes et `N` colonnes, et initialise toutes les cases de la matrice à la valeur `value`. La fonction renvoie l'adresse en mémoire de la matrice nouvellement créée.

```
int** createMatrix (int M, int N, int value);
```

3 Procédure `printMatrix`

Écrivez une procédure `printMatrix` qui affiche une matrice `mat` possédant `M` lignes et `N` colonnes.

```
void printMatrix (int **mat, int M, int N);
```

Vous pouvez vous baser sur l'existence de la fonction suivante qui affiche un tableau `tab` de taille `N`.

```
void printTab (int *tab, int N);
```

4 Fonction `getMatrixMaxNeg`

Écrivez une fonction `getMatrixMaxNeg` qui renvoie la plus grande valeur négative d'une matrice `mat` de `M` lignes et `N` colonnes.

```
int getMatrixMaxNeg (int **mat, int M, int N);
```

Vous pouvez vous baser sur l'existence de la fonction suivante qui renvoie la plus grande valeur négative d'un tableau `tab` de taille `N`.

```
int getTabMaxNeg (int *tab, int N);
```