

Allocation dynamique de tableaux à plusieurs dimensions

1. Allocation dynamique d'une matrice : Puissance NxM

Vous connaissez tous le jeu Puissance 4¹. Nous vous proposons de coder un jeu de puissance HxL. Ce jeu est en tout point identique au puissance 4 sauf qu'au lieu d'être joué sur un tableau de 6 cases de hauteur et de 7 cases de largeur, il se joue sur un tableau de H cases de hauteur et L cases de largeur.

Le programme vous demandera d'abord les valeurs H et L. Ensuite il vous demandera la colonne que joue le joueur X puis la colonne que joue le joueur O, puis celle du joueur X, et ainsi de suite.

A chaque coup, le programme affichera le tableau avec le caractère '.' pour les cases vides.

Dans un premier temps, nous ne vérifierons pas si l'un des joueurs a gagné. La partie se termine simplement lorsque l'un des joueurs entre la valeur 0.

Pour être « user friendly », nous commencerons à numéroté les colonnes à partir de 1 dans l'interface utilisateur (input/output) !

Vous allouerez dynamiquement la matrice en utilisant un tableau de tableaux (càd. des pointeurs de pointeurs). Allouez exactement la place nécessaire à la grille.

Exemple d'exécution :

```
Hauteur = ? 4
Largeur = ? 5

Colonne joueur X ? 3
. . . . .
. . . . .
. . . . .
. . x . .
Colonne joueur O ? 2
. . . . .
. . . . .
. . . . .
. o x . .
Colonne joueur X ? 3
. . . . .
. . . . .
. . x . .
. o x . .
Colonne joueur O ? 3
. . . . .
. . o . .
. . x . .
. o x . .
```

¹ https://fr.wikipedia.org/wiki/Puissance_4

2. Libération / réallocation de la mémoire

Reprenons le même jeu. Cette fois-ci les joueurs possèdent le pouvoir d'agrandir le tableau de jeu en rajoutant des colonnes. Il leur suffit pour cela de jouer un numéro de colonne plus grand que le nombre de colonnes courant.

Exemple :

```
. . . . .
. . O . .
. . x . .
. O x . .
Colonne joueur O ? 7
. . . . .
. . O . .
. . x . .
. O x . . O
```

Les colonnes seront rajoutées à l'aide de la fonction [realloc](#).

Si le joueur entre un nombre négatif, le nombre de colonnes est réduit. Par exemple, -3 ne conservera que les 3 premières colonnes.

```
. . . . .
. . O . .
. . x . .
. O x . . O
Colonne joueur O ? -3
. . .
. . O
. . x
. O x
```

Il faudra bien entendu libérer la mémoire de ces colonnes et réajuster la taille du tableau.

Bonus

Détectez le joueur gagnant ! (en supposant que H et L sont supérieurs à 4).

Exercice de debugging

3. Utilisation de l'outil *valgrind*²

Reprenez vos solutions des exercices 2 et 3 du TP3 ainsi que la solution de l'exercice précédent et vérifiez-les avec l'outil *valgrind*. Si une fuite de mémoire (*memleak*) ou toute autre erreur est détectée, corrigez-la.

² Voir document « Valgrind_tuto.pdf » sur Moovin.