

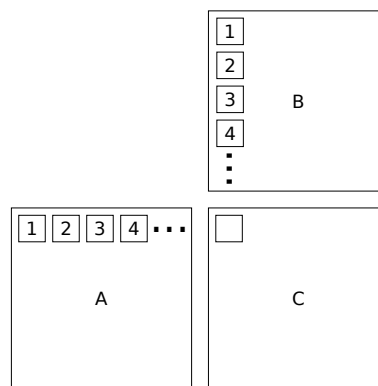
# SIMD TP2

## Multiplication de matrices

La liste des intrinsics SSE/AVX est disponible à l'adresse suivante : <https://software.intel.com/sites/landingpage/IntrinsicsGuide/>

### 1 Version scalaire

Multiplier 2 matrices A et B consiste pour chaque élément  $[i, j]$  de la matrice résultat C à calculer le produit scalaire de la ligne i de la matrice A par la colonne j de la matrice B :

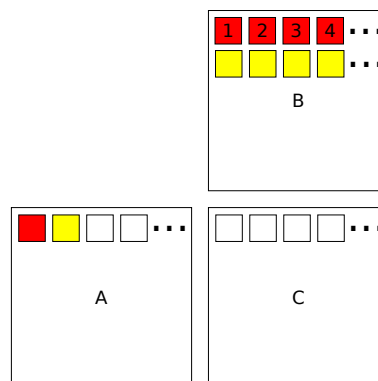


Écrire l'algorithme naïf de multiplication de matrices.

### 2 Version scalaire 2

La version ci-dessus n'est pas directement vectorisable car on accède aux éléments de la matrice B par colonne dont les données ne sont pas contiguës en mémoire donc impossibles à charger efficacement de manière vectorielle.

Pour pouvoir vectoriser efficacement, il faut partir d'un algorithme scalaire parcourant les matrices dans l'ordre ci-dessous :



Écrire le code correspondant. Utilisez votre algorithme naïf pour valider la correction de votre nouvel algorithme. Aidez vous du fichier `matmul.c`.

### 3 Version vectorielle

Pour simplifier les versions vectorisées, on suppose que la dimension de la matrice est un multiple de la taille des blocs que vous manipulez.

Écrire des versions vectorisées de la multiplication de matrices en utilisant les intrinsics SSE. Faites varier votre implémentation pour des blocs de tailles 4, 8 et 16.

Les intrinsics à utiliser sont `_mm_load_ps`, `_mm_store_ps`, `_mm_set1_ps`, `_mm_add_ps` et `_mm_mul_ps`.

1. Comparer les performances entre les différentes versions pour une matrice de taille 2048x2048 :
  - Quel est le gain apporté par la réorganisation des boucles ?
  - Quel est le gain apporté par la vectorisation SSE ?
  - Les gains sont-ils ceux attendus ?

### 4 Bonus - Version vectorielle par bloc

Il est aussi possible de travailler en multipliant des sous-blocs de la matrices plutôt que un à un. Simplement : on reprend l'un des algos précédemment définis, mais plutôt que travailler sur les éléments un à un, on prend des sous-blocs carrés dans la matrices.

Implémentez cette version et comparez avec vos précédentes versions.