

Série 7: Tableaux dynamiques avec `vector`

Lien avec le [MOOC Initiation à la Programmation \(en C++\)](#)

Lien avec ICC-Théorie en complément du MOOC

La suite de la partie théorique offre un peu moins d'opportunités d'illustrations en C++. Cela est compensé par le lancement du projet cette semaine. A ce propos, la donnée du projet inclut cette année la section « **2. Méthode de travail** » qu'il est bon de lire très rapidement. Lisez aussi en priorité la section « **5. Rendus** » pour être conscient de certaines contraintes comme la taille max des fonctions. La donnée est décomposée en 5 ACTIONS ; un [planning](#) cohérent avec le cours est indiqué sur le forum projet.

Exercices semaine5 du MOOC

- Document [Tutoriel « moyenne de classe »](#)
 - Remplissage d'un tableau au fur et à mesure de la lecture des données car on ne connaît pas a priori la taille qu'aura le tableau dynamique.
- Document [Exercices semaine 5 du MOOC](#)
 - **Exercice 15 : échauffement avec des tableaux dynamiques (niveau 1)**
 - Ajout d'un élément avec `push_back`
 - **Exercice 16 : produit scalaire (niveau 1)**
 - Passage par valeur de 2 tableaux `vector` à une fonction
 - **Exercice 17 : multiplication de matrices (niveau 2)**
 - Manipulation de tableaux `vector` à deux indices
 - Passage par référence de 2 tableaux à une fonction qui renvoie la valeur d'un tableau `vector`
 - Les arcanes de la *move semantics* ne sont pas au programme de notre cours
- Document [Exercices additionnels semaine 5 du MOOC](#)
 - **Exercice 14 : crible d'Erathostène (niveau 2)**
 - Cet algorithme a été mis au point il y a environ 2300 ans. La [visualisation de son fonctionnement](#) est très bien faite sur wikipedia. On peut aussi bien utiliser un `vector` ou un `array` de booléens car la taille est connue à l'écriture.
- Document [Tableaux « à la C »](#)
 - Lire ce document complémentaire avec les éléments de la classe inversée pour se faire une idée de l'organisation des données en mémoire. En particulier, dans le cas d'un tableau à deux indices l'organisation « ligne par ligne » permet de comprendre le coût constant de l'accès à un élément d'un tel tableau.