

# Révisions

Runtrack C++ Jour 1 : Un peu de révision

## Contexte

---

Dans le royaume de CodeTopia, une puissante équipe nommée "**La TeamPlat**" règne en maîtresse sur toutes les connaissances et compétences en programmation. Cependant, un maléfique bug appelé "**Errorus Fatalis**" menace de corrompre l'équipe, plongeant Codetopia dans le chaos. Seuls les plus braves et les plus astucieux programmeurs peuvent lever cette menace.

Votre mission, si vous l'acceptez, est de relever une série de défis en C++ pour restaurer l'ordre et la stabilité dans Codetopia. Chaque tâche est une étape vers la victoire, où vous apprendrez et démontrerez vos compétences de codage pour devenir le légendaire Maître du Code de Codetopia et rejoindre l'équipe. Préparez votre environnement de développement, aiguisiez votre esprit, et embarquez dans cette quête héroïque avec les exercices suivants.





## Job 0 : Préparation au combat

---

Avant de partir à l'aventure, il est crucial de s'assurer que votre épée (compilateur) est bien aiguisée. En fonction de votre système d'exploitation, suivez les instructions pour installer GCC :

- **Linux** : GCC devrait être déjà présent. Si ce n'est pas le cas, l'installer avec votre package manager (apt, pacman...)
- **Mac** : GCC (ou clang, qui est une bonne alternative) devrait être déjà présent. Si ce n'est pas le cas, installer GCC avec votre package manager (brew)
- **Windows** : Suivre le guide d'installation de MSYS2, y compris l'installation de GCC, puis ajouter le dossier bin/ de MSYS2 contenant les exécutables de GCC à votre variable d'environnement système Path.

Utiliser la commande "**g++ -v**" pour vérifier que le compilateur est bien installé et présent dans la variable d'environnement PATH.

## Job 1 : Le Salut des Nouveaux Héros

---

Écrire un programme en C++ pour afficher le message "**Hello world!**". C'est votre cri de guerre, le signal que vous êtes prêt à commencer l'aventure.



## Job 2 : L'Épreuve de la Multiplication

---

Un héros doit résoudre des énigmes. Écrire un programme en C++ permettant de **multiplier** deux nombres et d'**afficher leur produit**.

## Job 3 : La Sagesse des cinq

---



Testons votre capacité à gérer les ressources avec sagesse.

Écrire un programme en C++ qui **demande** à l'utilisateur de taper **cinq entiers** et qui affiche leur moyenne. **Vous êtes limités à deux variables**.

## Job 4 : Le duel Pair-Impair

---

Les héros doivent souvent déterminer la nature des choses.

Écrire un programme en C++ qui permet d'**afficher** si un nombre entier saisi au clavier est **pair** ou **impair**.

## Job 5 : L'échange des Destinées

---

Écrire un programme en C++ qui permet d'**échanger** le contenu de deux entiers **n** et **m** saisis par l'utilisateur, puis **afficher** ces entiers avant et après l'échange.



## Job 6 : La quête des Prix Justes

---



Vous êtes marchand de fruits et légumes dans l'agora de Codetopia. C'est un espace ouvert où les citoyens se rassemblent pour échanger des nouvelles, faire des affaires et discuter des affaires publiques. Aider un marchand de fruits en créant un programme qui **affiche** le prix TTC des produits frais, qui

**demande** à l'utilisateur de taper le prix HT d'un kilo de carottes, le nombre de kilos de carottes, et le taux de TVA (ex : 15 %). Le programme doit afficher le prix TTC d'une marchandise donnée.

## Job 7 : L'Inversion Mystique

---

Votre mission dans cette épreuve est de recevoir un nombre de la part de l'utilisateur, puis de manier habilement les chiffres pour en **inverser l'ordre**. Comme un sorcier des nombres, vous transformerez par exemple le nombre 12345 en 54321, démontrant ainsi votre expertise dans la manipulation des cycles et des mystères des nombres.



## Job 8 : Le Jugement de l'Année

---

Dans cette épreuve, votre tâche est de recevoir une année de la part de l'utilisateur et de discerner avec précision si elle est **bissextile**. Comme un oracle du temps, vous illuminerez votre chemin à travers les âges en déterminant si des années comme 2020, 2024 et 2028 sont bissextiles, ou si des années comme 1900, 2100 et 2200 ne le sont pas. Montrez votre habileté à naviguer dans les méandres du calendrier, apportant clarté et perspicacité à travers les époques.



## Job 9 : L'épreuve du Héros

---



Pour passer l'épreuve du Héros, écrire une fonction qui prend en entrée un **tableau d'entiers** et **sa taille**, puis **trouve** et **renvoie** le plus petit élément du tableau en utilisant des **pointeurs**.

## Job 10 : La Métamorphose du Héros

---

**Maintenant testons votre agilité !**

Modifiez la fonction précédente pour qu'elle prenne un tableau de pointeurs d'entiers au lieu d'un tableau d'entiers.



## Job 11 : Le Pacte des échanges

---

Écrire une fonction **swapValues** qui, tel un défi d'adresse, teste votre habilité à échanger avec maîtrise les valeurs de deux entiers en utilisant les références. Dans cette épreuve, vous êtes appelé à démontrer votre dextérité et votre efficacité. **Vous ne devez pas utiliser de pointeurs pour cet exercice.**

## Job 12 : La Magie du doublon

---

Écrire une fonction **doubleArray** qui prend un **tableau d'entiers** et **sa taille** en entrée, qui **double** chaque élément du tableau en utilisant des **pointeurs** pour accéder aux éléments du tableau, et **renvoie** la somme de tous les éléments doublés.

Ensuite, écrivez une fonction **printArray** qui prend un tableau d'entiers et sa taille en entrée, et **affiche** les éléments du tableau après les avoir doublés.

## Job 13 : L'Énigme des Fibonacci

---



La suite de Fibonacci est une séquence d'entiers où chaque terme est la somme des deux termes précédents.

Créer une fonction **fibonacci** qui calcule le **N**-ième terme de la suite de Fibonacci. Le résultat doit être calculé par le compilateur lors du traitement du code, et non lors de son exécution.



Après avoir écrit votre fonction, utilisez-la dans le programme principal pour afficher les dix premiers termes de la suite de Fibonacci. Assurez-vous que les valeurs sont correctes et que le programme fonctionne comme prévu.

Écrire ensuite une fonction qui prend un entier non-signé sur 32 bits en entrée et retourne **vrai** si le nombre de bits à 1 est pair et **faux** sinon. Testons ta capacité à démêler les mystères de cette suite et d'anticiper ses séquences numériques.

## Job 14 : Le Calcul du Cube

---

Écrire un programme qui demande à l'utilisateur de taper un entier **N** et qui **calcule** la somme des cubes de **5^3** à **N^3**.

## Job 15 : La quête des Bitiums

---

La quête des Bitiums consiste à retourner **vrai** si le nombre de bit est pair, et **faux** sinon. Par exemple, si nous avons un nombre binaire tel que 101010 (qui représente 42 en décimal), il y a trois bits définis à 1. Dans ce cas, le nombre de bits à 1 est impair et la fonction retournera **faux**. Pour cette épreuve, utilisez uniquement les opérations sur les bits disponibles en C++. Il n'est pas autorisé d'utiliser les opérateurs de comparaisons (==, != ...).

**Seriez-vous capable de relever le défi ?**

## Job 16 : Le Trius des Corda

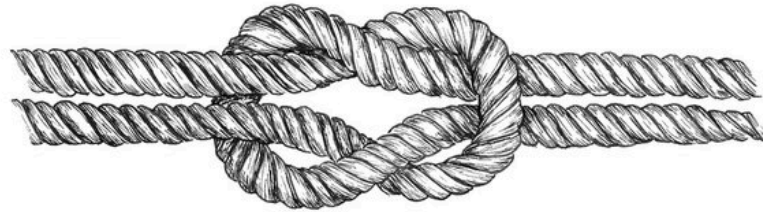
---

Écrire une fonction **MySortString** qui prend en entrée un tableau de chaînes de caractères et le **trie** en utilisant des **pointeurs**. Vous devez implémenter





votre propre algorithme de tri en utilisant des pointeurs pour accéder aux éléments du tableau et effectuer les échanges nécessaires. Démontrer votre maîtrise des algorithmes de tri.



## Job 17 : Le Colisée des Spécialités

---



Créer un programme pour une école qui gère un ensemble de spécialités.

Déclarer une énumération Spécialité avec cinq éléments : IA, Web, Logiciel, Système Immersif Et Cybersécurité. Utiliser un tableau pour stocker les spécialités des étudiants de la promo. Initialiser ce tableau avec les spécialités et

associer le nombre d'étudiants de manière aléatoire (entre 12 et 66). Écrire une fonction **afficherSpecialite** qui prend une spécialité en paramètre et affiche son nom ainsi que le nombre de places disponible. Dans le programme principal, parcourez le tableau de spécialités et utilisez la fonction "**afficherSpecialite**" pour afficher le nom de chaque spécialité.





## Job 18 : La Persévérance d'Héraclès



Imaginez-vous défier les épreuves divines comme l'aurait fait Héraclès dans la Grèce antique !

Dans cette quête : créer une fonction **addElement** qui prend en paramètre un tableau d'entiers, la taille actuelle du tableau et l'entier à ajouter. Cette fonction doit:

- allouer dynamiquement un nouveau tableau avec une taille augmentée de 1,
- copier les éléments de l'ancien tableau dans le nouveau,
- ajouter le nouvel entier à la fin du nouveau tableau,
- enfin, libérer la mémoire de l'ancien tableau.

Créer une fonction **showTable** qui prend en paramètre un tableau d'entiers et sa taille, et qui affiche tous les éléments du tableau.

Créer un tableau d'entiers vide avec une capacité initiale de 5 éléments. Ajoutez plusieurs entiers au tableau à l'aide de la fonction **addElement**, puis affichez le tableau à l'aide de la fonction **showTable**.

**Avez-vous réussi à surmonter cette épreuve comme l'aurait fait Héraclès ?**





## Compétences visées

---

→ C++

## Rendu

---

La runtrack doit être disponible sur votre github, au nom "**runtrack-cpp**".  
Les fichiers doivent être organisés précisément, comme indiqué dans les énoncés, dans un dossier correspondant à leur jour respectif.

**Exemple : jour01/job01/hello\_world.cpp**

## Base de connaissances

---

- [Apprendre le c++](#)
- [La syntaxe de base](#)
- [Aide mémoire C++](#)
- [Les boucles](#)
- [Les tableaux](#)
- [Les pointeurs](#)
- [Les pointeurs et références](#)