

**RENAUT**

**Langage C**

**Lucas**

**TP A**

**“CHAMPION DES MATHS”**

## **Table des matières**

I/ Introduction .....	3
II/ Fonctionnalités implémentes .....	4
III/ Difficultés rencontrées et solutions apportées.....	7
IV/ Commentaires et suggestions .....	8
V/ Auto-évaluation .....	10
VI/ Lien GitHub et structure du code .....	11

## **I/ Introduction**

### **DESCRIPTION :**

Le but de ce projet est de créer un programme qui permet au CM1 de s'entraîner sur des jeux de mathématiques de niveaux difficile comme l'addition, la multiplication, les aires, ou bien plus encore. Avec un système de points fonctionnel qui est sauvegarder si vous utiliser le même nom d'utilisateur.

Ce projet à un but pédagogique pour aider des élèves de CM1 à s'amuser tout en faisant des mathématiques.

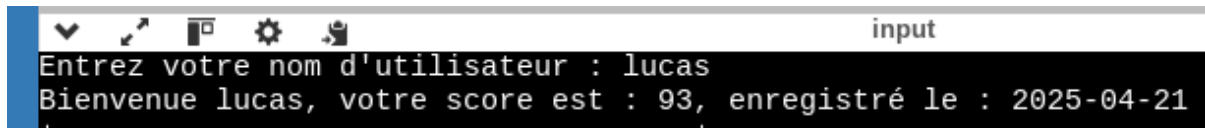
### **TECHNOLOGIES Utilisé :**

Pour réaliser ce projet nous avons utiliser

- Online GDB : pour exécuter le programme en ligne plus rapide, mais moins sur
- Code: :blocks : pour exécuter le programme localement, plus long mais plus sur
- Langage de programmation : **C**

## **II/ Fonctionnalités implémentes**

### **Fonctionnalité 1 : sauvegarde et chargement des points**

A screenshot of a terminal window with a dark background. The title bar at the top shows standard window controls and the word 'input'. The terminal text is as follows:

```
Entrez votre nom d'utilisateur : lucas
Bienvenue lucas, votre score est : 93, enregistré le : 2025-04-21
```

#### **Rôle et utilité :**

Ces fonctionnalités permettent de personnaliser l'expérience utilisateur. Chaque joueur peut retrouver son score d'une session précédente. Cela donne un sentiment de progression et rend le jeu plus motivant.

#### **Fonctionnement :**

Lorsqu'un joueur saisit son nom, le programme utilise la fonction `charger_score` pour lire le fichier `scores.txt` et chercher si une entrée correspond à son nom. Si oui, il récupère le score. Sinon, il commence à 0. À la fin de la session de jeu du joueur, `sauvegarder_score` met à jour ou crée une nouvelle ligne avec le nom du joueur, son score et la date.

#### **Problèmes rencontrés et solutions :**

L'un des défis a été la gestion du format de ligne lors de la lecture et l'écriture dans le fichier. Il fallait être très précis dans le `sscanf` pour extraire correctement le nom, le score et la date, en gérant les espaces et les virgules.

2° la mise à jour d'une ligne existante nécessitait de passer par un fichier temporaire, car on ne peut pas modifier une ligne directement dans un fichier texte. On a utilisé un fichier temporaire pour écrire les nouvelles données, puis j'ai remplacé `scores.txt` par celui-ci.

#### **Fonctionnalité mise en avant :**

La gestion de score est visible dès le démarrage du jeu et à la fin. Elle donne une continuité au programme entre plusieurs lancements.

#### **Choix techniques :**

Utilisation de fichiers texte pour la simplicité. J'ai utilisé `fopen` en mode `r+` et `w` pour manipuler le fichier selon qu'on le lit ou qu'on le recrée. L'emploi de `strftime` permet d'enregistrer une date claire et lisible, ce qui rend le fichier plus compréhensible pour le joueur.

## **Fonctionnalité 2 : Menu principale du jeu**

```
+-----+
|1 : Addition
|2 : Soustraction
|3 : Multiplication
|4 : Tables des multiplications
|5 : Divisions
|6 : perimetre
|7 : aire
|8 : melange de calcul
|0 : Sortir du jeu
+-----+
Quel est votre choix ?
9

+-----+
|1 : Addition
|2 : Soustraction
|3 : Multiplication
|4 : Tables des multiplications
|5 : Divisions
|6 : perimetre
|7 : aire
|8 : melange de calcul
|0 : Sortir du jeu
+-----+
Quel est votre choix ?
4
```

### **Rôle et utilité :**

Cette fonctionnalité permet au joueur de naviguer sur le jeu qu'il a envie de faire pour lui permettre de s'entraîner sur les outils mathématiques où cette élève rencontre des difficultés. Créant ainsi une sensation de plaisir car il n'est pas forcé.

### **Fonctionnement :**

À l'aide du "scanf", l'utilisateur saisit le numéro correspondant à l'exercice que l'élève souhaite faire. Le programme utilise un switch pour rediriger l'utilisateur vers la fonction appropriée. Le menu est ensuite rappelé à chaque fois qu'une opération est terminée, ce qui permet de relancer un exercice sans redémarrer le programme.

### **Problèmes rencontrés et solutions :**

Un problème fréquent a été la gestion incorrecte des entrées utilisateur. Par exemple, si l'utilisateur saisisait un caractère au lieu d'un nombre, cela pouvait faire planter la boucle ou l'amener à un comportement inattendu. Pour limiter cela, on a restreint le menu de manière qu'il ne repose que sur des entiers en utilisant " (scanf("%d", &choix))" (le "%d" permet de définir les entiers) Pour palier à cela on a mis une boucle while pour garder le menu actif jusqu'à ce que l'utilisateur entre un entier compris dans le switch.

### **Fonctionnalité mise en avant :**

Le menu rend visible toutes les fonctionnalités du jeu en un seul endroit, ce qui améliore considérablement l'expérience utilisateur. Il agit aussi comme un système de navigation simple et efficace.

### **Choix techniques :**

On a choisi de faire un choix simple pour éviter la complexité d'une interface graphique. En utilisant La boucle while avec switch-case garantit que le code est facilement extensible (ajouter une nouvelle option est simple).

## **Fonctionnalité 3 : Mécanismes de points et tentatives multiples**

```
multiplication
10*8= ?
Entrez le résultat :
vous avez entrez 8
retentez votre chance
Entrez le résultat :
vous avez entrez 23
derniere chance
Entrez le résultat :
vous avez entrez 80
bravo
votre score est de :94
```

### **Rôle et utilité :**

Le système de points permet d'évaluer la performance du joueur, tout en lui laissant plusieurs chances. Cela est important pour ne pas décourager les élèves tout en

récompensant ceux qui réussissent du premier coup. Ce système est utilisé dans toutes les fonctions des exercices (addition, soustraction, etc....).

### **Fonctionnement :**

Chaque exercice commence par une question. Si l'utilisateur trouve la bonne réponse dès le premier coup, il obtient 10 points. S'il se trompe, il a droit à deux autres tentatives, pour 5 points puis 1 point respectivement (le système est un peu différent pour les tables de multiplications car il obtient 3 points s'il réussit du premier coup, puis il a encore 2 tentatives pour 2 points puis 1 point respectivement). Ce mécanisme est programmé avec des conditions "If-else" imbriquées et utilise la fonction `comptage()` pour ajouter les points.

### **Problèmes rencontrés et solutions :**

L'un des pièges était de bien s'assurer que chaque tentative affiche correctement la question et que le score soit mis à jour uniquement une fois par tentative. Une autre difficulté a été d'éviter les répétitions de code. J'ai contourné cela en gardant la logique similaire dans toutes les fonctions, même si cela aurait pu être amélioré.

### **Fonctionnalité mise en avant :**

Visible dans chaque type de calcul, ce système rend l'apprentissage interactif, adaptable au niveau de l'utilisateur et permet au joueur de voir ces erreurs tout en apprenant.

### **Choix techniques :**

Utilisation de `scanf` répétitifs avec messages adaptés pour chaque essai. La fonction `comptage` agit comme un compteur global qui centralise la gestion du score.

## **III/ Difficultés rencontrées et solutions apportées**

Lors de la réalisation de mon projet, j'ai rencontré plusieurs difficultés, en particulier lors de la conception du fichier. Voici un détail des problèmes rencontrés ainsi que les solutions que j'ai mises en place pour y remédier.

### **Difficulté 1 : La conception du fichier**

L'un des principaux problèmes que j'ai rencontrés lors de la conception du projet est le système de chargement du score, je ne savais pas réellement comment faire pour reconnaître le nom entre avec un nom qui est dans un fichier appart

Pour surmonter ce problème, j'ai utilisé plusieurs méthodes :

- **Demande d'aide au professeur :** Premièrement j'ai décidé de demander à mon chargé de TP s'il serait possible d'obtenir des indications sur comment lire ligne par ligne et ce qu'il a fait en m'expliquant certaine notion comme le « sscanf » ou « strcmp » et ainsi a réussi à me débloquent
- **Visionnage d'un tutoriel :** cela a pu confirmer mes hypothèses et à plus m'éclaircir sur la forme générale pour le chargement du fichier et m'a également aidé pour importer dans le fichier d'enregistrement la date et l'heure.

### **Autres difficultés**

On a eu d'autre difficulté mais qui sont mineurs et qui sont en réalité plus des erreurs d'étourderie comment oublier un « & » et oublier le « \* » pour le pointeur, cela créant des erreurs dans le programme et des incohérences. Mais ses problèmes on les a réglé de deux manières :

- **demande d'aide au professeur :** on a demandé de l'aide au prof pour permettre de connaître le type d'erreurs et le plus souvent ce sont des petites erreurs d'étourderie.
- **Travail de groupe :** Dans le groupe on s'entraidait pour essayer de trouver les petites erreurs et cela évite de demander au chargé de TP, déplus cela nous permet d'apprendre.

## **IV/ Commentaires et suggestions**

### **1° Ce qui fonctionne bien**

- Chargement et sauvegarde des scores

Le mécanisme d'ouverture/création du fichier scores.txt est fiable et permet de récupérer son score à tout moment.

- Interface utilisateur textuelle claire

Les menus sont lisibles et sont très simple à comprendre, les demandes de saisie explicites et l'élèves sait toujours où il en est (choix de l'exercice, nombre de chances, etc.).

- Variété des exercices



Choix entre addition, soustraction, multiplication, divisions, calculs mixtes... le panel d'exercices couvre l'essentiel du programme de CE1 de maths, malgres que ce soit un niveau assez dure les élèves ont de quoi travaillées

## **1. Points à améliorer**

### **Duplication de code**

Beaucoup de ligne de commande qui sont dupliquer (gestion des trois chances, affichage du score) sont répétés dans chaque fonction. Cela nuit à la stabilités .

### **Vérification de la saisie**

Une entrée non numérique si le menu principal du jeu fait planter le programme. Il serait utile de vérifier le retour de scanf et de voir si l'élève a entre une chaîne de caractère ou un nombre.

## **2. Suggestion d'amélioration**

### **Ajout de niveaux de difficulté**

Proposer un niveau "facile / moyen / difficile" qui adapte la taille des nombres ou le nombre de décimales demandées pour que l'élève puisse s'y retrouver s'il a un niveau moyen ou à peut de se rater si c'est trop difficile.

### **Statistiques et historique**

À la fin de chaque session, afficher un bilan (score total, taux de réussite moyen par type d'exercice, progression par rapport à la session précédente).

### **Mode chronométré**

Limiter le temps de réponse pour rendre l'exercice plus dynamique et ajouter un bonus de rapidité par exemple et si l'élève ne répond pas assez rapidement il ne gagne aucun point.

## **3. Apprentissage**

### **L'importance des fonctions :**

Découper le programme en fonctions pour chaque type d'exercice. Permet d'avoir un programme beaucoup plus concis et plus léger

## **CHAMPION DES MATHS - TPA**

### **La gestion des fichiers en C :**

Fopen en mode lecture/écriture, création de fichiers temporaires, remove/rename. ,  
Les principales commandes d'extraction de fichier et comment retrouver un utilisateur

### **Tester chaque fonction indépendamment :**

Il vaut mieux tester une fonction indépendamment pour éviter les erreurs et donc de ne pas savoir où l'erreurs (dans la fonction ou le code principale)

## **V/ Auto-évaluation**

ASPECT	NOTE (sur 10)	Description
Codage	6,5	On a effectué le travail qui nous a été demandé sur certain point on aurait pu être plus dans les détails et rajouter d'autre fonctionnalité citée dans la partie IV , de plus le code pourrait contenir des bogues que nous n'avions jamais vue
Travail de groupe	8	On a bien travaillé dans le groupe, il y avait de l'entre aide entre nous, quand un membre de groupe avait besoin on était là pour lui répondre
Planification	7	On bien gérer le temps que nous avons eu droit pour ce TP on aurait pus optimiser encore plus le temps .

## **CHAMPION DES MATHS - TPA**

Autonomie	5	On a travaillé chez nous mais je pense que le travail personnel aurait pus être grandement amélioré quand même
Robustesse	5	Notre code a beaucoup de code qui est dupliqué (système de chance) ce qui pourrait être optimisé de plus il y a aucune vérification de saisie dans le menu

**Note globale selon moi pour la partie codage : 6,3/10**

### **Objectifs pour les projets futurs**

Le but est clairement d'améliorer la structure générale et les vérifications du code en générale car c'est le problème majeur de notre code ici, avec un manque aussi de documentation dans le code en lui-même ce qui peut porter à confusion si une personne emprunte notre code pour l'améliorer, et rajouter plus de fonctionnalité dans le code pour permettre plus de diversité et un plus grand plaisir pour l'utilisateur qu'il soit élèves ou personne lambda

## **VI/ Lien GitHub et structure du code**

### **Voici un lien pour accéder au GIT HUB :**

<https://github.com/lucasrenaut/RENAUT-Lucas---TP-A>

