

Jeux Genetec

Problème d'optimisation

# 1 Introduction

Vous êtes maintenant en dernière année de baccalauréat. Pour votre projet d'intégration, on vous demande de travailler en équipe de 4 sur un projet de moyenne à grande envergure. Pendant la phase de planification du projet, vous avez établi et défini des modules qui ont une interface spécifique, ce qui vous permet de développer plus rapidement et d'appliquer une méthodologie de programmation que vous avez vu dans l'un de vos cours, soit le Test Driven Development (TDD). Vous utilisez donc vos interfaces pour faire de l'injection de dépendance et vous utilisez des mocks pour tester individuellement vos modules.

Il arrive maintenant le temps de finaliser votre projet qui est dû la semaine prochaine et vous vous rendez compte que le module d'encodage de votre collègue est sous-optimal. En fait, le code est simplement affreux. Puisque la période commence sous peu et que vous n'avez toujours pas étudié, votre temps est une denrée rare. Vous déterminez qu'avec 3h, vous devriez avoir assez de temps pour améliorer son code pour qu'il soit beaucoup plus rapide tout en demeurant fonctionnel. Votre ami se propose également de vous aider.

À vous de jouer !

## 2 Requis

On vous demande d'optimiser un projet qui lit un certain nombre de fichiers et qui les encode en base64. Voici les critères dont vous devez respecter :

1. Il est **interdit** de modifier le `cxproj`, la solution, ou toute autre option de compilation (excepté les command lines arguments, afin de pouvoir spécifier vos chemins pour vos fichiers de tests).
2. Le programme doit être compilé en **debug**.
3. Tous les fichiers qui se retrouvent dans le répertoire d'output (2<sup>ème</sup> paramètre passé au programme) seront corrigés.
4. Les résultats produits par votre programme doivent être **identiques** à ceux qui seraient générés par le programme d'origine.
5. Chaque ligne du fichier "Statistique.txt" doit être ordonnée **alphabétiquement** selon le nom de fichier.
6. L'exécutable « FileGen.exe » fourni est le même qui sera utilisé pour générer les fichiers lors de la correction.
7. Le temps d'exécution calculé dans le `OptimisationProblem.cpp` ne sera pas utilisé lors de la correction. Un outil externe est responsable de calculer le temps d'exécution total de votre programme lors de la correction.
8. Vous devez remettre un fichier explicatif « Explications.txt » qui détaille les optimisations que vous avez apportées au programme. Une explication d'une ligne par optimisation est suffisante. Exemple :

OptimisationProblem.cpp  
fonction main(): changer la création d'objet A en string, évite *new* sur le *heap*

## 3 Correction

Puisque le code est fonctionnel, il est crucial que votre code le soit encore. Si vos résultats sont incorrects, vous serez automatiquement disqualifié. Par la suite, nous évaluerons l'équipe dont leur programme est le plus rapide.

## 4 Remise

Vous devez remettre un zip (nom1-nom2.zip) qui contient les fichiers suivants :

```
+ OptimisationProblem
|--+ OptimisationProblem
|   |-- Directory.cpp
|   |-- Directory.h
|   |-- Encoder.cpp
|   |-- Encoder.h
|   |-- File.cpp
|   |-- File.h
|   |-- FileReader.cpp
|   |-- FileReader.h
|   |-- FileWriter.cpp
|   |-- FileWriter.h
|   |-- HelperMethods.h
|   |-- Logger.cpp
|   |-- Logger.h
|   |-- OptimisationProblem.cpp
|   |-- OptimisationProblem.vcxproj
|   |-- OptimisationProblem.vcxproj.filters
|   |-- Statistics.cpp
|   |-- Statistics.h
|   |-- stdafx.cpp
|   |-- stdafx.h
|-- OptimisationProblem.sln
|-- Explications.txt
```

L'adresse de remise des fichiers est au <https://dropitto.me/JeuxGenetecEpreuve2>.

Nous bloquerons le dépôt une fois la période de dépôt terminée.

La durée de cette épreuve est de 3 heures.