

DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

**Structures de données et algorithmes**  
**Devoir 4 — Énoncé**

---

SIGLE : INF21307  
TITRE : Structures de données et algorithmes  
GROUPE : 06  
PROFESSEUR : Steven Pigeon  
K-212  
steven\_pigeon@uqar.ca

MODALITÉ : À faire seul ou en équipe de deux.  
DATE DE REMISE : 29 avril 2022, avant minuit.

---

Total : **15 pts.** (Langage C++, Linux) Grâce au code fourni, développez votre fonction de hachage.

1. **5 pts.** Bricolez votre propre fonction de hachage. Le prototype de la fonction doit correspondre, exactement, au prototype de la fonction déjà fournie :

```
hash_uint_t pourrie(const hash_key & hk)
```

où `hash_key` est une classe qui contient un pointeur vers les données à hacher et la longueur de ces données (cela permet, grâce à une fonction avec une spécialisation explicite, d'appliquer la même fonction de hachage à à peu près n'importe quoi, pas seulement des `std::strings`).

2. **5 pts.** Votre fonction de hachage doit être aussi uniforme que possible.
3. **2 pts.** Votre fonction doit bien couvrir les 64 bits de la valeur retournée.

4. **3 pts.** Votre fonction doit être indépendante de la longueur de la clef (les clefs courtes ne reçoivent donc pas nécessairement de petits *hashes*, ni les longues les gros).

Votre fonction *doit être originale* : vous ne pouvez pas reprendre du code qui traîne sur Internet, ni un morceau de code d'une librairie existante. Vous devez fournir votre propre code et une fonction de hachage originale.

Le code fournit permet de tester automatiquement votre fonction de hachage avec un fichier (lu ligne par ligne) :

```
test-hash 100 < mots.txt > histo.txt
```

Va lire du fichier `mots.txt` et produire l'histogramme de 100 entrées dans le fichier `histo.txt`. Évidemment, vous pouvez choisir ce que vous voulez pour la taille de l'histogramme.

Le code et les données pour le tester se trouvent sur Moodle.

\*  
\* \*

La qualité du français écrit est de mise, et jusqu'à 10% de la note finale pourraient être retranchés. N'oubliez pas de citer vos sources, mêmes celles dont vous vous inspirez plus ou moins vaguement. Enfin, vous vous exposez à 20% de pénalité par jour de retard, jusqu'à concurrence de quatre jours — sauf circonstances *exceptionnelles*.

Pour la remise du devoir, préparez une archive (zip, 7z, etc.) avec votre solution *avec le code au complet* (pas besoin de réinclure les données, je les ai déjà), et dont le nom sera `votrenom.devoir4.zip`. Cette archive sera déposée sur Moodle.