**LOG 2810**

**STRUCTURES DISCRETES**

Rapport de laboratoire #2

TP2 : Automates

|  |  |
| --- | --- |
| **Éléments évalués** | Points |
| **Qualité du rapport** : respect des exigences du rapport, qualité de la présentation des solutions | 2 |
| **Qualité du programme** : compilation, structures de données, gestion adéquate des variables et de toute ressource (création, utilisation, libération), passage d’arguments, gestion des erreurs, documentation du code, etc. | 3 |
| **Composants implémentés**: respect des requis, logique de développement, etc. |  |
| C1 | 3 |
| C2 | 3 |
| C4 | 2 |
| C5 | 3 |
| C6 | 2 |
| Total de points | 20 |

Soumis par :

Bouchard, Billy - 1850477

Gnaga, Dogbeda Georges - 1870143

Obossou Ema-Wo, Sanyan - 1780896

Le 12 Avril 2018

Table des matières

[Introduction 3](#_Toc508229100)

1. Présentation……………………………………………………………………………………………………………………………………….3

1.1 Fichier automate.py……………………………………………………………………………………………………………..4

1.1 Fichier solution.py…………………………………………………………………………………………………………..…..6

1.1 Fichier main.py…………………………………………………………………………………………………………………...6

[2. Difficultés rencontrées 7](#_Toc508229101)

[3. Solutions aux difficultés rencontrées 7](#_Toc508229101)

[Conclusion 8](#_Toc508229100)

# **Introduction**

## 

Dans l’optique de mettre en pratique des notions de base sur la théorie des graphes apprises durant le cours de Structures discrètes, nous avons implanté une application pour trouver le chemin optimal afin de réaliser un braquage dans une ville et de fuir dans une autre ville.

Dans l’optique de mettre en application les notions théoriques sur les automates apprises durant le cours de structures discrètes, nous sommes amenées à créer une application qui grâce à des automates et banques de mots génère des mots de passe.

L’objectif ici est d’aider un étudiant cambrioleur d’avoir accès aux coffres-forts de ses banques cibles.

Pour ce faire nous avons à partir de quelques fichiers contenant des règles linguistiques, de fichiers contenant des variantes correspondantes à ces règles là et surtout grâces à des automates qui reconnaissent tous les mots dont ces règles définissent, trouver les mots de passes pour accéder à ces coffres-forts.

Nous avons implémenté une fonction qui sert à créer un automate, une fonction qui permet de trouver les mots de passe, une fonction qui gère les fichiers impliquer et une dernière fonction qui affiche les mots de passe trouvés.

1. **Présentation**
   1. **Fichier automate.py**

Ce fichier est composé de :

Not finish

Figure 1 : Diagramme de classe complet de l’application.

* 1. **Fichier solution.py**
  2. **Fichier main.py**

# **Difficultés rencontrées**

Voici les difficultés auxquelles nous nous sommes confrontés :

* Recherche régulière des diverses syntaxes pour écrire nos algorithmes.
* Débogage du code un peu plus complexe vu la maîtrise partielle du langage.
* Difficulté pour maitriser l’algorithme qui implémente les automates.

1. **Solutions aux difficultés rencontrées**

Nous nous sommes imposé une bonne discipline dans la réalisation de notre application, et nous avons opté pour les solutions pour nous en sortir telles que :

* Visionnement de plusieurs tutoriels sur le fonctionnement des automates.
* Révision des notions théoriques sur les automates vus en classe.
* L’entraide collective dans notre équipe nous a permis de facilement repérer nos erreurs et de les corrigés.
* Tests unitaires réguliers et continu des méthodes et fonctions durant les implantations.

## 

**Conclusion**

Not finish