

Etape 1 :

Développer un programme C++ permettant de charger un automate d'états finis (Finite State Machine).

Utiliser une `map<pair<int,char>,int>` pour stocker les transitions.

Question : Pourquoi cette structure ne permet de stocker que des automates déterministes ??
Quelle est la structure qui permet de stocker des automates non-déterministe ??

Etape 2 :

Développer un algorithme (une méthode pour la classe automate) permettant de vérifier si un mot appartient au langage de l'automate.

Charger à partir de Teams les fichiers **automate.cpp** et **automate.txt**.

Chaque ligne représente une transition dans l'automate sous la forme :

Etat1 Caractère Etat2

Qui signifie qu'avec l'état 1 et le caractère de l'alphabet on va à l'état 2.
Les états finaux sont représentés par un entier négatif.

Etape 3 : Exercice ouvert

Ecrire un programme permettant de classer une chaîne de caractères saisie par l'utilisateur (identificateur, constante réelle ou naturelle, constante binaire ou hexadécimale, inconnu) :

1. En utilisant plusieurs automates.
2. En utilisant un seul automate.