Cours 420-202-RE Traitement de données orienté objet Hiver 2021 Cégep Limoilou Département d'Informatique	Tp 5 mandat 3 de 4 (1 semaine pour cette partie) 12% pour les 4 mandats Cryptographie - Le chiffre de Hill

OBJECTIFS

- Utiliser et manipuler des structures de données sur disque et en mémoire;
- Trouver une solution informatique à un problème;
- Utiliser les méthodes appropriées des classes de l'API de Java;
- Livrer un code documenté et testé.

ACTIVITÉ À RÉALISER

- Voir le fichier du mandat 1.

CONTRAINTE :

- Voir le fichier du mandat 1.

MANDAT 3 :

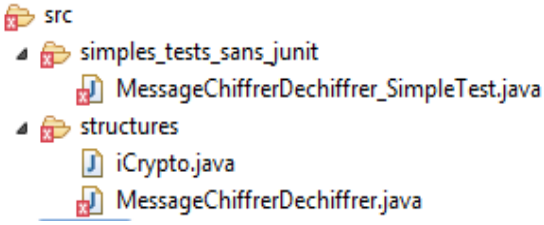
Documentation et recherche :

- Voir le fichier du mandat 1.
- Voir les notes de cours et les exercices sur la récursivité.

Gestion du chiffrement et du déchiffrement :

- Avant de commencer le mandat 3, il faut terminer et bien tester tous les éléments du mandat 2.
- Maintenant que vous avez développé les utilitaires et structures de données qui forment un « framework » complet pour la suite de votre laboratoire, vous pouvez copier le code du mandat 3, disponible sur le réseau, et l'intégrer à votre projet des mandats 1 et 2.
- À partir du code fourni sur le réseau vous devez, selon la « JavaDoc » fournie, compléter les différentes méthodes de la classe « MessageChiffrerDechiffrer ».
    - Dans le package « structures » vous avez la nouvelle classe « MessageChiffrerDechiffrer » qui implémente l'interface « iCrypto » et qui gère le chiffrement (codage) et déchiffrement (décodage) de Hill, d'un message à partir d'un vecteur de caractères, d'une liste de matrices candidates et d'un ensemble de mots (dictionnaire) reçus lors de la construction d'un objet « MessageChiffrerDechiffrer ».
    - Il faut aussi produire une classe de tests JUnit pour la classe que vous allez compléter dans ce mandat. Il est conseillé de produire les tests au fur et à mesure que vous développez vos méthodes. **Testez correctement car c'est la base de votre solution globale, vous devez avoir confiance en votre code.**
- À partir du code fourni sur le réseau vous trouverez aussi, dans le package « simples_tests_sans_junit », une nouvelle classe qui utilise de façon sommaire la classe développée dans ce mandat. J'ai mis en commentaire la sortie que j'ai obtenue à partir de mon code et des valeurs d'origine qui sont dans la classe de tests. **Ces tests sont simplement fournis pour valider vos sorties avec les miennes. Il ne faut pas considérer ces tests comme étant suffisants pour dire que tout fonctionne correctement.**

- Voici les différents packages et classes qui vous sont fournis pour ce mandat.



Échéancier :

- Vous avez une semaine pour réaliser le travail demandé par ce mandat.
- Il n'y a rien à remettre pour ce mandat, vous devez simplement avoir terminé et testé correctement votre code avant de recevoir le prochain mandat.

Chiffrement de Hill

Encodage :

Exemple **mat**rice choisie 3x3 : Message : Salut les amis 14 de longueur (donc + 1 espace)

1	3	4
5	6	7
8	9	10

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	~~b~~	-

(sal)

mat[0][0] * i de s + mat[0][1] * i de a + mat[0][2] * i de l = 1 * 18 + 3 * 0 + 4 * 11 = 62 % 28 = 6 →G

mat[1][0] * i de s + mat[1][1] * i de a + mat[1][2] * i de l = 5 * 18 + 6 * 0 + 7 * 11 = 167 % 28 = 27 →-

mat[2][0] * i de s + mat[2][1] * i de a + mat[2][2] * i de l = 8 * 18 + 9 * 0 + 10 * 11 = 254 % 28 = 2 →C

(ut~~b~~)

mat[0][0] * i de u + mat[0][1] * i de t + mat[0][2] * i de esp. = 1 * 20 + 3 * 19 + 4 * 26 = 181 % 28 = 13 →N

mat[1][0] * i de u + mat[1][1] * i de t + mat[1][2] * i de esp. = 5 * 20 + 6 * 19 + 7 * 26 = 396 % 28 = 4 →E

mat[2][0] * i de u + mat[2][1] * i de t + mat[2][2] * i de esp. = 8 * 20 + 9 * 19 + 10 * 26 = 591 % 28 = 3 →D

idem avec (les) → LJY

(~~b~~am) → SSU

(is~~b~~) → ~~b~~WK Ce qui fait donc : G-CNEDLJYSSU~~b~~WK

Décodage : On essaie les différentes matrices candidates

1ere)

1	2	3
4	5	6
7	8	10

Matrice inverse de Hill :

18	8	1
18	13	26
1	26	1

Message à décoder : G-CNEDLJYSSU~~b~~WK

(G-C)

mat[0][0] * i de G + mat[0][1] * i de - + mat[0][2] * i de C = 18 * 6 + 8 * 27 + 1 * 2 = 326 % 28 = 18 →S

mat[1][0] * i de G + mat[1][1] * i de - + mat[1][2] * i de C = 18 * 6 + 13 * 27 + 26 * 2 = 511 % 28 = 7 →H

mat[2][0] * i de G + mat[2][1] * i de - + mat[2][2] * i de C = 1 * 6 + 26 * 27 + 1 * 2 = 710 % 28 = 10 →K

etc. pour (NED), (LJY), (SSU), (~~b~~WK) ce qui donne SHKRAIOPRMOCKGU → pas bon

2e)

1	2	3
4	5	6
7	9	10

Matrice inverse de Hill :

8	21	27
10	15	2
19	11	27

(G-C)

mat[0][0] * i de G + mat[0][1] * i de - + mat[0][2] * i de C = 8 * 6 + 21 * 27 + 27 * 2 = 669 % 28 = 25 →Z

mat[1][0] * i de G + mat[1][1] * i de - + mat[1][2] * i de C = 10 * 6 + 15 * 27 + 2 * 2 = 469 % 28 = 21 →V

mat[2][0] * i de G + mat[2][1] * i de - + mat[2][2] * i de C = 19 * 6 + 11 * 27 + 27 * 2 = 465 % 28 = 17 →R

etc. pour (NED), (LJY), (SSU), (~~b~~WK) ce qui donne ZVRRAIBNE~~b~~OQQW → pas bon

3e)

1	2	4
5	6	7
8	9	10

Matrice inverse de Hill :

1	4	22
26	26	5
1	7	20

Donne SALLJRPYW~~b~~AM~~b~~KQ → pas bon

Finalement 4e)

1	3	4
5	6	7
8	9	10

Matrice inverse de Hill :

27	2	27
2	2	23
27	5	25

Donne SALUT~~b~~LES~~b~~AMIS~~b~~ → trouvé !!