

Question 7 :

Lors du remplissage de la table, si un hash est déjà présent dans la table (sans le **t**), la fin de la chaîne sera la même.

Avec le paramètre **t**, on ajoute *un peu d'aléatoire* dans le calcul de la chaîne.

Question 8

```
./main --create 100 200 table
```

Question 9

```
./main --info table 10
```

Question 10

~~

Question 11

La recherche dans une table arc-en-ciel à une complexité de $O(\text{largeur}^2 * \text{hauteur})$ dans le pire cas car si toute la table est candidate alors on doit toute la parcourir.

Question 12

~~

Question 13

On n'est pas arrivé à retrouver le texte clair à partir d'un hash directement.

Par contre si on passe le texte clair on arrive à le hasher puis à le retrouver.

```
./main --alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ --min-size 4 --max-size 4 --create 1000  
100000 table
```

```
./main --alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ --min-size 4 --max-size 4 --crack ABCD  
table --> Found it ABCD
```

Question 14

~~

Question 15

~~

Question 16

Le CLI n'est pas exactement comme le votre et comporte seulement les fonctions qui fonctionnent correctement.

Pour voir la liste de commandes, écrivez `./main -h` ou `./main --help`

Note personnelles

Pour voir toutes les commandes docker que j'ai fait

```
$ cat  
C:\Users\comco\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost  
_history.txt | findstr "docker"
```

Pour lancer le docker avec un alpine

```
$ cd C:\Users\comco\Documents\FAC YOU\M2\info1002\TP  
$ docker rm info1002-container  
$ docker run -it --name info1002-container -v ${pwd}:/app frolovlad/alpine-gcc  
$ docker start info1002-container  
$ docker exec -it info1002-container sh
```

Dans le docker

```
$ apk add make openssl libressl-dev g++
```

Liens

```
https://github.com/CLIUtils/CLI11
```