Imerzoukene Anisse G-04
Elouali Wissam G-04

TP CRYPTOGRAPHIE Partie 2 :

12) La classe est publique puisqu'on veut pouvoir accéder à la clé publique toutefois on va rendre privée la partie privée de la clé privée.

```
public  -> e
public  -> n
private -> d
```

13) - e et z doivent être étrangers.
- d et z doivent être étrangers.
- p et q doivent être étrangers
- On doit vérifier que (d * e) % z est égal à 1.

```
if (!(Premier.Etranger(e, z) && Premier.Etranger(d, z) && ((d * e) % z == 1) && Premier.Etranger(p, q)))
```

14) d = e ^ φ(z) [z]

15) Nous avons réalisé une méthode permettant de déterminer un e au hasard.
On vérifie donc uniquement les entrées p et q puis on détermine un e au hasard.

```
RSA r = new RSA(13, 71);
Console.WriteLine($"{r.toString()}");
```

```
RSA avec e déterminé au hasard :
e : 223 n : 923 d : 727
```

```
RSA avec e déterminé au hasard :
e : 83 n : 923 d : 587
```

On a d'ailleurs dans la liste ePossible de la méthode CalculClee, tous les e possiblement utilisables d'où la levé d'une exception si aucun e n'est possible.

Mais nous avons également réalisé la méthode classique.

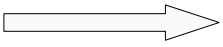
```
RSA R1 = new RSA(5,17,5);
Console.WriteLine(R1.toString());
```

```
FLORENCE
e : 5 n : 85 d : 13
```

```
RSA R1 = new RSA(4,11,3);
Console.WriteLine(R1.toString());
```

```
throw new Exception("Paramètres invalides :");
```

```
RSA R1 = new RSA(3,11,40);
Console.WriteLine(R1.toString());
```



```
throw new Exception("Paramètres invalides :");
```

17 & 19) Jeu de tests avec (p, q) = (47, 71) et e déterminé au hasard.

```
e : 1217 n : 3337 d : 3093
Val pre chiffrement : 61
Val post chiffrement : 1322
Val dechiffre avec cle privée connu : 61
Val dechiffre sans cle : 61
```

```
e : 1733 n : 3337 d : 877
Val pre chiffrement : 61
Val post chiffrement : 2263
Val dechiffre avec cle privée connu : 61
Val dechiffre sans cle : 61
```

```
e : 281 n : 3337 d : 2521
Val pre chiffrement : 651
Val post chiffrement : 1006
Val dechiffre avec cle privée connu : 651
Val dechiffre sans cle : 651
```

21) On pourrait utiliser le codage ASCII étendu. En regardant la table ASCII de 0 à 255 en décimal, comme l'alphabet une valeur entre 0 et 255 correspondrait à un caractère.

22) Cette méthode est hors RSA, car nous n'avons pas besoin des clés pour réaliser cette tâche. On est ici dans un autre univers, on pourrait créer une classe dédiée à cet univers ASCII ou Alphabet mais étant donné que les futurs méthodes que nous créerons se baseront sur l'alphabet, intégrer ces méthodes dans la classe RSA aussi semble une bonne idée.

23) Argument 1 : On sait que le chiffrement et déchiffrement passe par l'exponentiation qui est une méthode assez lourde. On veut donc réduire le nombre de blocs à chiffrer.

Argument 2 : Le même entier sera toujours chiffré de la même manière, on ne veut pas que notre chaîne soit en proie à l'analyse fréquentielle.

25) Jeu de tests :

```
e : 839 n : 3337 d : 1539
Question 27 : Exemple de chiffrage d'une chaine :
220
675
2031
2957
2814
2438
Question 29 : Exemple de dechiffrage d'une chaine :
FLORENCE
```

```
e : 67 n : 3337 d : 2403
Question 27 : Exemple de chiffrage d'une chaine :
2544
2352
58
1060
1665
2464
```

28) On retrouve bien le même cryptogramme pour « FLORENCE ».

```
Question 27 :
17011435264511112604442
1701
1435
2645
1111
2604
442
Question 28 :
61
215
180
514
30
500
FLORENCE
```

30) Autre exemple avec la chaîne « JADORE ».

```
RSA avec e déterminé au hasard :
e : 137 n : 923 d : 233
```

```
Question 27 : Exemple de chiffrage d'une chaine avec e au hasard:
225011068213596
225
1
10
682
135
96
Question 28 : Exemple de dechiffrage d'une chaine avec e au hasard:
10
1
4
15
18
5
JADORE
```

Merci pour votre lecture.