Laboratoire 14

- **Buts**
  - Implantation de fonctions simples
- **Travail à réaliser**
  - Implanter une fonction permettant de déterminer si un nombre entier fourni en paramètre est premier
  - Implanter la fonction d'exponentiation modulaire $b^e \mod m$, où b, e et m sont des entiers positifs. Pour implanter cette fonction efficacement, on peut remarquer que si e est pair, sa valeur vaut $((b^2) \mod m)^{(e/2)}$, ce qui permet de diviser par 2 le nombre de multiplications. Si b est impair, sa valeur vaut : $b \cdot b^{e-1} \mod m$. On en dérive l'algorithme efficace donné ci-dessous, à implanter sous la forme d'une fonction.
  - Implanter une fonction probabiliste permettant si un nombre entier fourni en paramètre est premier. On utilisera l'algorithme de test de primalité rapide donné ci-dessous

  Écrire un petit programme permettant de tester et de vérifier le comportement de ces fonctions
- **Délai**
  - Fin de la semaine

Algorithme d'exponentiation modulaire

Input: $b, e, m \in \mathbb{N}$
Result: $r = b^e \mod m$

```
1 r ← 1
2 while e > 0 do
3 |   if e mod 2 = 0 then
4 |   |   b ← b^2 mod m; e ← e/2
5 |   else
6 |   |   r ← r · b mod m; e ← e − 1
7 end
```

Test rapide de primalité

Propriétés d'un nombre p premier :

- $\forall a < p, a^{p-1} \equiv 1 \mod p$
- 1 n'a que 2 racines carrées modulo p : (1 et $p-1$)

Input: $p \in \mathbb{N}$
Output: `false` si p non premier ; `true` si p probablement premier

```
1  if p < 2 then return false;
2  if p = 2 then return true;
3  begin Répéter 10 fois
4  |   Générer un nombre aléatoire a < p
5  |   if a^{p-1} ≠ 1 mod p then return false;
6  |   q = 1; u = p − 1
7  |   while u pair and q = 1 do
8  |   |   u ← u/2
9  |   |   q ← a^u mod p
10 |   |   if q ≠ 1 and q ≠ p − 1 then return false;
11 |   end
12 end
13 return true
```