L'objet de ce TP est de générer de grands nombres premiers. La recherche de grand nombres premiers se fait en deux étapes : **trouver un nombre** susceptible d'être premier puis **tester** s'il est vraiment premier.

Trouver un nombre Pseudo-premier

Pour ce faire on utilise une des formules suivantes :

1. **Expression affine-Dirichlet :** Il existe une infinité de nombres premiers de la forme $6n + m$ où n et m sont premiers entre eux.
   - Écrire la fonction `Dirichlet` qui admet en paramètres d'entrée un entier N puis retourne N nombres de la forme $6k + 1$.
   - Écrire la fonction `Dirichlet2` qui admet en paramètres d'entrée deux entiers p et N puis retourne N nombres de la forme $30k + p$ avec p premier.
2. **Polynôme du second degré :** Pour chacun des cas suivant, écrire une fonction générant N nombres selon le polynôme donnée.
   - Polynôme **d'Euler** $n^2 + n + 41$.
   - Polynôme **de Legendre** $2n^2 + 29$.
   - Polynôme **de Ruby** $103n^2 - 3945n + 3431$.

Test de Fermat

Les algorithmes permettant de déterminer si un nombre est premier sont des algorithmes très lents. En général on teste le nombre avec un algorithme probabiliste beaucoup plus rapide qui indique si le nombre est composé ou très probablement premier.

Parmi ces tests, celui de Fermat s'appuie sur le petit théorème de Fermat : Si p est premier $a^{p-1} \equiv 1[p]$.

Si un nombre p ne vérifie pas l'égalité alors il n'est pas premier. S'il vérifie l'égalité il est assez probable qu'il soit premier (la réciproque du théorème n'est pas vraie). Un nombre p passant le test avec les bases $a = 2$,3,5 et 7 a une forte porbabilité d'être premier.

1. Écrire une fonction vérifiant les test de Fermat avec comme base $a = 2$.
2. Compléter la fonction pour tester le nombre avec les bases 2,3,5 et 7.
3. Trouver deux grands nombres pseudo-premiers.
4. Calculer les pourcentage de nombres pseudo-premiers obtenus à l'aide des polynômes d'Euler pour des chiffres inférieurs à 100000. En théorie l'ordinateur donne un nombre premier dans 47,5% des cas pour des chiffres inférieurs à 10 millions.
5. Comparer les pourcentage de nombres pseudo-premiers obtenus avec les différentes formules définies dans la première partie.

Test de Rabin-Miller

Le test de Rabin-Miller est un raffinement du test de Fermat. En plus d'exploiter le petit théorème de Fermat, il exploite cette propriété spécifique aux nombres premiers qui suit: il est toujours vrai que

$$1^2 = -1^2 = 1 \mod n$$

pour tout entier n, premier ou non. Ce qui, par contre, est vrai seulement pour les nombres premiers p c'est que -1 et 1 sont les deux seules racines carrées de 1. Par exemple on a

$$1^2 = -1^2 = 4^2 = 11^2 = 1 \mod 15.$$

Étant donné un entier n à tester, le test de Rabin-Miller procède comme suit:

On écrit $n - 1 = 2^s d$ avec d impair;

On choisit $1 < a < n - 1$ au hasard;

On calcule $a^d \mod n$, si c'est égal à 1 ou -1 on ne peut rien dire et on arrête;

Pour i allant de 1 jusqu'à s on calcule $a^{d2^i} \mod n$:

si $a^{d2^i} = -1 \mod n$ on ne peut rien dire et on arrête;

si $a^{d2^i} = 1 \mod n$ on a trouvé une racine carrée de l'unité différente de 1 et -1, on conclut que n est composé et on arrête;

Si on est arrivés à la fin de la boucle et que $a^{d2^s} \neq 1 \mod n$, on conclut que n est composé, comme d'après le test de Fermat.

Le test de Rabin-Miller est très efficace: on peut montrer que pour tout n la probabilité qu'un a pris au hasard soit un menteur est au plus 1/4 (et souvent en pratique beaucoup plus faible). Si on répète k fois le test de Rabin-Miller (avec des a différents au hasard) la probabilité d'identifier erronément un nombre composé comme étant premier est bornée par $1/4^k$.

Dans les applications pratiques on estime souvent qu'une vingtaine de répétitions du test de Rabin-Miller suffisent à affirmer avec confiance qu'un nombre est premier.

Implémenter une fonction `test_Rabin` qui permet de tester si un nombre n est premier.