Exercice 1 Il s'agit de définir des **prédicats** (fonctions renvoyant un booléen). Pour chacun :

- Définir quels sont ses paramètres et contraintes d'utilisation, s'il y en a.
- Rédiger une docstring complète.
- Ecrire le code de la fonction.
- Valider ce code à l'aide des doctests.
- Choisissez un nom de fonction adapté (voir question **1.**).

Proposez des prédicats pour vérifier :

1. si un nombre passé en paramètre est impair

```
>>> est_impair(13)
True
```

```
>>> est_impair(8)
False
```

2. si un nombre passé en paramètre est multiple de 7
3. si une chaîne de caractères est vide (proposez deux codes différents)
4. si un nombre est positif
5. si deux nombres sont tous deux positifs
6. si deux nombres sont de même signe
7. si deux nombres sont de signes opposés
8. si un nombre est dans un intervalle donné (bornes comprises).

Exercice 2:Etats de l'eau À une pression atmosphérique normale, l'eau est à l'état solide lorsque la température est inférieure à 0°C, à l'état gazeux au delà de 100°C, et à l'état liquide sinon. La température ne peut atteindre la valeur -273.15°C et les valeurs inférieures.

Proposez un prédicat `est_glace` pour vérifier si l'eau est sous forme de glace à une température donnée en paramètre.

Proposez un prédicat `est_liquide` pour vérifier si l'eau est liquide à une température donnée.

Exercice 3 Dans le programme suivant a, b, c sont les mesures **entières** des côtés d'un triangle.
N'hésitez pas à vous servir d'un interpréteur Python pour répondre aux questions.

1. Quels sont les paramètres de cette fonctions ?
2. Quel est le *type* de la variable d ? de la variable t ? du retour de la fonction ?
3. Analyser ce que renvoie cette fonction.

```
def f(a, b, c):
    d = "non équilatéral"
    if a==b and a==c:
        d = "équilatéral"
    s = a+b+c
    t = a < (b+c)
    return d,s,t
```

Exercice 4 Le pourcentage de français ayant les yeux bleus est de 8 %.

1. Ecrire une fonction `yeux_bleus` qui renvoie un booléen indiquant si une personne choisie au hasard a les yeux bleux. Quels sont les paramètres de cette fonction ?
2. Ecrire une fonction `nb_bleus` qui utilise la fonction précédente et renvoie le nombre de personnes ayant les yeux bleus dans un échantillon aléatoire de taille n. Quels sont les paramètres de cette fonctions ?

Indication: Pour simuler le tirage aléatoire d'une personne dans une population de 100 personnes, utiliser le module `random`.

Exercice 5 Une forêt de 30 000 *ha* perd 8 % sa population chaque année. Programmer une fonction `nb_annee` prenant la surface de la forêt en paramètre qui renvoie, si on laisse cette situation catastrophique en l'état:

1. la taille de cette forêt 10 ans plus tard ? `nb` ans plus tard ?
2. le nombre d'années pour que cette forêt ait perdu la moitié de sa surface.

Exercice 6 Dans cet exercice a, b et c sont des entiers.

1. Ecrire une fonction `somme_carres` qui renvoie la somme des carrés de deux entiers `a` et `b`.
2. Ecrire une fonction `est_pythagore` qui renvoie un booléen indiquant si les réels `a, b, c` peuvent être les mesures des côtés d'un triangle rectangle d'un triangle d'hypoténuse c.
3. Ecrire une fonction `pythagore` qui renvoie un booléen indiquant si `a, b, c` peuvent être les les mesures des côtes d'un triangle rectangle.

Exercice 7 :Nombres premiers Ecrire une fonction `est_premier` prenant en paramètre un nombre entier m et retournant `True` si ce nombre est premier, `False` sinon. Ce type de fonction s'appelle un **prédicat**. Un nombre premier est nombre entier non nul qui n'admet que 1 et lui-même comme diviseur.
Exemples:

```
>>>est_premier(100)
False
```

```
>>>est_premier(17)
True
```

 : Les nombres premiers sont très présents en cryptologie moderne. Le système RSA notamment les utilise. Son algorithme basé sur une propriété simple des nombres premiers sert notamment à sécuriser les transactions par internet. On estime que plus de 300 millions de programmes installés peuvent utiliser le RSA.

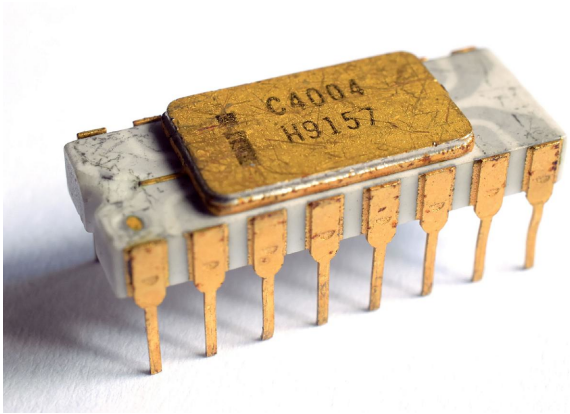

Figure 1: 1971-Le 4004, premier microprocesseur

Exercice 8: Algorithme en C++ On imagine un jeu avec une raquette et une balle dans le style de l'historique *Pong*. Lorsque le jeu est lancé, l'ordinateur attend un *évènement*: utilisation du clavier ou de la souris par le joueur). `niveau` est le niveau qui est en train d'être joué.

Pong est sorti en 1972 sur borne d'arcade

Pong sort en 1975 sur Atari VCS. La sortie et le succès de ce jeu sur la console d'Atari est considéré comme l'évènement précurseur au lancement de l'industrie du jeu vidéo, avec une forte augmentation de l'offre, des centaines de consoles de salon reprenant le concept.

Il faut associer les débuts de l'industrie du jeu vidéo à la commercialisation du premier microprocesseur 4004 d'Intel en novembre 1971. Ce dernier a permis la création et production en masse des consoles de jeux.

Expliquer ce que peuvent faire les fonctions suivantes, effectuer des recherches si nécessaire.

1. ```
   void play_one_game()
   {
       while (!game_over){
           attente_evt();
           mise_a_jour();
           dessiner();
       }
   ```

2. ```
   int set_score(int score, short niveau, short max)
   {
       short bonus = 1, i = 0;
       do
       {
           score += bonus;
           bonus *= 2;
           i ++;
       } while(bonus <= max || i <= niveau)
       return score;
   }
   ```

3. La variable `NB_BALLES_INITIAL` contient le nombre de balles au début du jeu. On déclare habituellement les **variables globales** en majuscule, ce qui explique pourquoi sa déclaration n'apparaît pas dans le corps de la fonction: ce n'est pas une **variable locale** à cette fonction.

   ```
   void play()
   {
       for (int i = NB_BALLES_INITIAL; i > 0;  i--)
       {
           play_one_game();
       }
   }
   ```

4. ```
   int set_score(int score, short niveau, short max)
   {
       short bonus = 1;
       for (i=0; i<=niveau;i++)
   ```
   ```

```
    {
        score += bonus;
        bonus *= 2;
        if (bonus > max)
        {
            break;
        }
    }
    return score;
}
```

Exercice 9: Algorithme d'Euclide L'algorithme d'Euclide permet de déterminer le plus grand diviseur commun (PGDC) entre deux entiers positifs A et B. Selon Donald Knuth, un des pionniers de l'algorithmique, c' est l'un des plus anciens algorithmes connu, on le retrouve en effet dans le livre VII des **éléments d'Euclide** vers 300 avant JC.

Pour obtenir ce PGDC, il suffit d'effectuer la division euclidienne de A par B, et de récupérer le reste r.

Puis A devient B et B devient r et on continue ainsi tant que ce reste n'est pas nul. Le pgdc de A et B est le dernier reste non nul.

1. Calculer à la main le PGDC de 21 et 15.
2. Ecrire une fonction `pgdc` renvoyant le PGDC de deux entiers positifs.

Exercice A: Approximation de la racine carrée L'algorithme du mathématicien grec Héron d'Alexandrie (1^{er} siècle) permet de calculer des valeurs approchées de $\sqrt{a}$ avec a réel positif:

- On choisit un nombre positif u.
- On calcule $\dfrac{u + \dfrac{a}{u}}{2}$
- On recommence en remplaçant u par la valeur trouvée à l'étape précédente et ainsi de suite.
- d'étape en étape la valeur de u se rapproche de $\sqrt{a}$ (on dit que la suite de nombre converge vers $\sqrt{a}$).

Remarque: Soit la suite (u_n) définie par:

$$\begin{cases} u_0 & = 1 \\ u_{n+1} & = \dfrac{u_n + \dfrac{a}{u_n}}{2} \end{cases}$$

(u_n) converge vers $\sqrt{a}$.

---

1. Calculer à la main $\sqrt{2}$ pour 5 itérations.
2. Ecrire une fonction `racine(a, n)` prenant deux entier `a` et `n` en paramètre et renvoyant une valeur approchée de $\sqrt{a}$ après n étapes de l'algorithme de Héron et la tester.
3. On souhaite améliorer l'algorithme en calculant les termes tant que la différence entre deux termes successifs est supérieure à une valeur `precision` donnée. Ecrire une nouvelle fonction `racine` prenant un nombre réel `a` et un nombre entier `precision` renvoyant la valeur approchée de $\sqrt{a}$. Pour rappel la distance entre deux nombres est définie par la valeur absolue $|a - p|$.

Exemples:

```
>>>racine(100)
10
```

```
>>>racine(10, 4)
3.1622
```

 : On dit que la limite de u_n quand n tend vers l'infini est égale à $\sqrt{a}$, ce qui se note $\lim\limits_{n\to+\infty} u_n = \sqrt{a}$

Exercice B: Le bon type de donnée ?

1. Donner la représentation en base 2 des décimaux 15 et 87.
2. Ecrire une fonction `dec2bin` prenant en paramètre un entier et renvoyant la liste de ses chiffres en binaire.
3. Ecrivez une fonction `dec2bin` renvoyant une chaine de caractère représentant le nombre en base 2.
4. Vérifier pour les décimaux 15 et 87.

Sources

1. Liste et Forum NSI.
2. Site de Didier Müller: `https://www.didiermuller.ch/index.php`

Figure 2: Pepper et Carrot par David Revoy CC-BY-SA 4.0