Exercice 1 Justifier la terminaison puis démontrer que la fonction suivante renvoie le produit de a par b par un invariant de boucle.

```
def multiplication(a,b):
            p = 0
            i = 0
        while i < a:
            p = p + b
            i = i + 1
          return p
```

Exercice 2 Justifier la terminaison puis démontrer que la fonction suivante renvoie la somme des valeurs de la liste L par un invariant de boucle.

```
def multiplication(L):
           s = 0
           i = 0
     while i < len(L):
          s = s + L[i]
           i = i + 1
          return s
```

On prendra pour convention que $\sum_{k=0}^{-1} a_k = 0$

Correction de l'exercice 1 Soit (p_k) et (i_k) les suites définies par $\begin{cases} p_0 = 0, i_0 = 0 \\ p_{k+1} = p_k + b \\ i_{k+1} = i_k + 1 \end{cases}$

- $(a - i_k)$ est une suite d'entiers positifs car $a \in \mathbb{N}$ et (i_k) est une suite arithmétique de raison 1 et de premier terme 0 tel que $i_k < a$.
  De plus $a - i_{k+1} = a - i_k - 1 < a - i_k$, cette suite est strictement décroissante.
  Elle est bien un variant de boucle.
  Le programme se termine.
- Soit, pour tout $k \in \{1, \dots, n\}$, l'assertion $\mathcal{P}(k) : p_k = i_k \times b$.
  Avant la boucle : Comme $i_0 = 0$ et $p_0 = 0$,
  Donc l'assertion $\mathcal{P}(0)$ est vraie.
  Pendant la boucle : Soit $k \in \{0, \dots, n-1\}$. Supposons que $\mathcal{P}(k)$ est vraie.
  Après la $(k+1)$-iéme itération on a :
  $$p_{k+1} = p_k + b = i_k \times b + b = (i_k + 1) \times b = i_{k+1} \times b$$
  Car $i_{k+1} = i_k + 1$.
  Donc l'assertion $\mathcal{P}(k+1)$ est vraie.
  A la fin de la boucle : Par récurrence finie, l'assertion est vraie pour $i_n = a$.
  On aura alors à la fin de la dernière boucle $p_n = a \times b$.
  C'est ce que l'on voulait obtenir.
  La fonction fait bien ce qu'on veut.

Correction de l'exercice 2 Soit (s_k) et (i_k) les suites définies par $\begin{cases} s_0 = 0, i_0 = 0 \\ s_{k+1} = s_k + L[i_k] \\ i_{k+1} = i_k + 1 \end{cases}$

- $(\texttt{len}(L) - i_k)$ est une suite d'entiers positifs car $\texttt{len}(L) \in \mathbb{N}$ et (i_k) est une suite arithmétique de raison 1 et de premier terme 0 tel que $i_k < \texttt{len}(L)$.
  De plus $a - i_{k+1} = a - i_k - 1 < a - i_k$, cette suite est strictement décroissante.
  Elle est bien un variant de boucle.

Le programme se termine.

– Soit, pour tout $k \in \{1, \dots, n\}$, l'assertion $\mathcal{P}(k) : s_k = \sum_{j=0}^{i_k-1} L[j]$.

Avant la boucle : Comme $i_0 = 0$ et $s_0 = 0$, et comme, $\sum_{j=0}^{0-1} L[j] = 0$, par convention, l'assertion $\mathcal{P}(0)$ est vraie.
Pendant la boucle : Soit $k \in \{0, \dots, n-1\}$. Supposons que $\mathcal{P}(k)$ est vraie.
Après la $(k+1)$-iéme itération on a :

$$s_{k+1} = s_k + L[i_k] = \sum_{j=0}^{i_k-1} L[j] + L[i_k] = \sum_{j=0}^{i_k+1} L[j]$$

Comme $i_{k+1} = i_k + 1$.
Donc l'assertion $\mathcal{P}(k+1)$ est vraie.
A la fin de la boucle : Par récurrence finie, l'assertion est vraie pour $i_k = \texttt{len}(L)$.
On aura alors à la fin de la dernière boucle $s_k = \sum_{j=0}^{\texttt{len}(L)-1} L[j]$.
C'est ce que l'on voulait obtenir.
La fonction fait bien ce qu'on veut.