Pour les exercices demandant de compléter un programme python les parties manquantes sont représentées par `?(1)?`, `?(2)?`, `?(3)?`... Le format de votre réponse sera par exemple

```
(1)   a=b
(2)   return x
(3)   f(x,y)
```

où `a=b` est le contenu manquant à l'endroit `?(1)?`,
`return x` est le contenu manquant à l'endroit `?(2)?`,
`f(x,y)` est le contenu manquant à l'endroit `?(3)?`...

Exercice 1 *Sachant que l'on dispose d'une fonction* `fus(tt1,tt2)` *qui retourne l'union triée de deux tableaux* `tt1,tt2` *triés,*

1. *Complétez ce programme de tri fusion en python*

```
def trif(t):
    if  len(t) <= 1:
        return t
    m = len(t)//2
    t1 = t[:m]
    t2 = t[m:]
    tt1= ?(1)?
    tt2= ?(2)?
    return ?(3)?
```

2. *Quelle équation de récurrence est satisfaite par $T(n)$ le temps d'exécution de* `trif` *sur un tableau de longueur n ?*
3. *Donnez les détails de la résolution de cette équation pour $n = 2^p$.*
4. *Donnez les détails de la résolution de cette équation pour tout n.*

Exercice 2 1. *Complétez ce programme de tri insertion en python.*

```
def trins(A):
        for j in range(1,len(A)):
                x=A[j]
                i=j-1
                while  ?(1)?
                         ?(2)?
                        i -= 1
                A[i+1]=x
```

2. *En notant* `A[:j]` *le sous-tableau* `A[0],A[1],...,A[j-1]`, *montrez que* `trins` *est valide.*
3. *Si* `A` *est trié dans l'ordre croissant combien y a-t-il d'exécution de* `i -= 1` *?*
4. *Si* `A` *est trié dans l'ordre décroissant combien y a-t-il d'exécution de* `i -= 1` *?*
5. *Quelle est la complexité de* `trins` *en moyenne ?*

Exercice 3 1. *Montrer que* `Euclide` *retourne le pgcd de a, b.*

```
def Euclide(a,b):
    """on suppose a>=b"""
    if b==0:
        return a
    else:
        return Euclide(b,a%b)
```

2. *Montrer que la complexité de* `Euclide` *est* $O(\log b)$.

Exercice 4 1. *Le pseudo-code suivant décrit un algorithme récursif pour la multiplication de deux matrices carrées:*

```
def mult(A,B):
     if n <= 1:
           return C=A*B
     else:
          C11=mult(A11,B11)+mult(A12,B21)
          C12=mult(A11,B12)+mult(A12,B22)
          C21=mult(A21,B11)+mult(A22,B21)
          C22=mult(A21,B12)+mult(A22,B22)
          return C
```

Expliquez ce pseudo-code.

2. *Quelle est la complexité de* `mult` *?*

3. *Sachant que*

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ + & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & + & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ + & + & \cdot & \cdot \end{pmatrix} + \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ + & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ - & \cdot & \cdot & \cdot \end{pmatrix}$$

$$\begin{pmatrix} \cdot & \cdot & + & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & + & + \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} + \begin{pmatrix} \cdot & \cdot & \cdot & - \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

et que

$$\begin{pmatrix} + & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & + \\ \cdot & - & \cdot & + \end{pmatrix} = \begin{pmatrix} + & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ + & \cdot & \cdot & + \end{pmatrix} + \begin{pmatrix} \cdot & \cdot & \cdot & - \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} - \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ + & + & \cdot & \cdot \end{pmatrix}$$

$$\begin{pmatrix} + & \cdot & - & \cdot \\ + & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & + \end{pmatrix} = \begin{pmatrix} + & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ + & \cdot & \cdot & + \end{pmatrix} + \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ + & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ - & \cdot & \cdot & \cdot \end{pmatrix} - \begin{pmatrix} \cdot & \cdot & + & + \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

donnez le pseudo-code d'un algorithme récursif pour la multiplication de deux matrices carrées A, B *de taille* n, *dont le temps d'éxécution satisfait* $T(n) = 7T(n/2) + \Theta(n^2)$.

4. *Donez un algorithme qui multiplie deux nombres complexes* $u = x + iy$ *et* $z = a + ib$ *en n'effectuant que 3 multiplications.*

Exercice 5 1. *Complétez cette fonction python pour qu'elle retourne un triplet* $(x, y, d) \in \mathbb{Z}^3$ *tels que* $ax + by = d$, *où* d *est le pgcd de* a, b.

```
def B(a,b):
        """on suppose a>=b"""
        if b==0:
                return [1,0,a]
        else:
                [x,y,d]=B(b,a%b)
                return ?(1)?
```

2. *Répondez à l'aide d'une fonction python à la question suivante : étant donnés le pgcd* d *de* a, b, *les couples d'entiers relatifs* (x, y) *tels que* $ax + by = d$ *sont-ils uniques ?*