Correction

Cor. 3.3 :

```
from math import sin,pi
def nombre_elements_positifs(l):
    k=0
    N=0
    while k<len(l):
        if l[k]>=0:
            N+=1
        k+=1
    return N
L=[sin(k+1) for k in range(100)]
print(nombre_elements_positifs(L))
```

Cor. 3.4 :

```
def plus_petit_element(l):
    if l==[]:
        return None
    m=l[0]
    K=0
    k=1
    while k<len(l):
        if l[k]<m:
            m=l[k]
            K=k
        k+=1
    return (m,K)
```

L'exécution de cette fonction sur les listes demandées donne
$(0.1411200080598672, 2)$ puis
$(0.008851309290403876, 21)$ et enfin
$(3.014435335948845e-05, 354)$.
On peut en déduire que $\pi \approx 3$ puis que $\pi \approx \frac{22}{7}$ et enfin que $\pi \approx \frac{355}{113}$ à 3.10^{-5} près.

Cor. 3.5 : On utilise les chaînes de caractères :

```
def f(a):
    return a+2*int(str(a)[-1::-1])
```

$f(17)$ renvoie 159. Pour les autres valeurs de a, on fait de même ou mieux, on utilise une boucle sur l'ensemble des valeurs données. On obtient respectivement 2379, 15429 et 232572. Tous ces nombres sont divisibles par 3.
On peut prouver que pour tout $a \in \mathbb{N}$, $f(a)$ est toujours divisible par 3. Comment ?

Cor. 3.6 :

```
def recherche(chaine,mot):
    i=0
    cpt=0
    while i<=len(chaine)-len(mot):
        if chaine[i:i+len(mot)]==mot:
            cpt+=1
```

```
        i+=1
    return cpt
```

Cor. 3.7 :

```
def recherchebis(chaine,mot):
    i=0
    cpt=0
    indices=[]
    while i<=len(chaine)-len(mot):
        if chaine[i:i+len(mot)]==mot:
            cpt+=1
            indices.append(i)
        i+=1
    return cpt,indices
```

Cor. 3.8 :

```
def remplace(chaine,mot1,mot2):
    i=0
    res=str()
    while i<=len(chaine)-len(mot1):
        if chaine[i:i+len(mot1)]==mot1:
            res=res+mot2
            i+=len(mot1)
        else:
            res=res+chaine[i]
            i+=1
    return res
```

Cor. 3.9 : `sum([k**4 for k in range(1,11)])`

Cor. 3.10 :

```
def somme(n,p):
    return sum([k**p for k in range(1,n+1)])
```

Cor. 3.11 :

```
def binomiaux(N):
    res=[[1]*(k+1) for k in range(N+1)]
    for k in range(2,N+1):
        for i in range(1,k):
            res[k][i]=res[k-1][i-1]+res[k-1][i]
    return res
```