

# Correction

*# Exercice 1*

```
def alterne(n):
    v=4
    for i in range(n):
        v=v+(-1)**(i+1)*4/(2*i+3)
    return v

print("alterne(1000) : ",alterne(1000),"\nà comparer avec pi :",np.pi)
print("Remarque : converge très lentement.")
print()
```

*# Exercice 2*

*# a-*

```
def Tchebychev(n,x):
    a,b=1,x
    for i in range(n):
        a,b=b,2*x*b-a
    return a

# b-
print("Tchebychev(50,np.cos(1)) :",Tchebychev(50,np.cos(1)))
print("à comparer avec np.cos(50) :",np.cos(50))
```

*# Exercice 3*

```
def listeCarac(chaine):
    res = []
    for c in chaine:
        if c not in res:
            res.append(c)
    return res

print(listeCarac("bonbon"))
```

*# Exercice 4*

```
def listeNombreCarac(chaine):
    res = []
    carac = []
    for c in chaine:
        if c not in carac:
            carac.append(c)
            res.append((c,1))
        else:
            i = carac.index(c)
            res[i] = (res[i][0], res[i][1]+1)
    return res

print(listeNombreCarac("bonbon"))
print(listeNombreCarac("oppose"))
```

### # Exercice 5

```
def memesLettres(mot1,mot2):
    lettres1 = listeCarac(mot1)
    lettres2 = listeCarac(mot2)
    for c in lettres1:
        if c not in lettres2:
            return False
    for c in lettres2:
        if c not in lettres1:
            return False
    return True
print(memesLettres("bonbon", "bon"))
print(memesLettres("oppose", "appose"))
```

### # Exercice 6

```
def anagrammes(mot1,mot2):
    lettres1 = listeNombreCarac(mot1)
    lettres2 = listeNombreCarac(mot2)
    for c in lettres1:
        if c not in lettres2:
            return False
    for c in lettres2:
        if c not in lettres1:
            return False
    return True
print(anagrammes("bonbon", "bon"))
print(anagrammes("chien", "chine"))
```

### # Exercice supplémentaire

```
def poly(n,z):
    L=[]
    for k in range(2**n):
        P=k%2
        m=1
        k=k//2
        while k>0:
            m*=z
            P+=(k%2)*m
            k=k//2
        L.append(P)
    return L

import matplotlib.pyplot as plt
n=12
z=(1+1j)/2 # essayer aussi avec z=(1+7**0.5*1j)/4 (par exemple)
for V in poly(n,z):
    plt.plot(V.real,V.imag,'.b')
plt.axis('scaled')
```