# Corrigé – Les chaînes de caractères

```
# Ex 1
s = "a b"*2 + "ba" + "b"*3
print(s, len(s)) # affiche 'a ba bbabbb' 11

##########################################################################

# Ex 2
# La fonction convert renvoie la même lettre en majuscule

##########################################################################

# Ex 3
def rang(l):
    return ord (l) - 97

def lettre(r):
    return chr(r + 97)

##########################################################################

# Ex 4
s = "abn\nn\\\""
print(len(s)) # affiche 7

##########################################################################

# Ex 5
s = "abcdefghij"
n = len(s)
print(s[1], s[5]) # affiche 'b' 'f'
s = s*2
print(n) # affiche 10
print(s[12]) # affiche 'c'

##########################################################################

# Ex 6
ch = "abcdef"
n = len(ch)
x = ch[n-1]
# ou
x = ch[len(ch)-1]
# ou
x = ch[-1]

##########################################################################
```

```python
# Ex 7

def est_numerique(car):
    pt_code = ord(car)
    if pt_code >= 48 and pt_code <= 57:
        return True
    return False

def est_numerique(car):
    return ord(car) >= 48 and ord(car) <=57

res1 = est_numerique("k")
print(res1)
res2 = est_numerique("4")
print(res2)

##########################################################################

# Ex 8
def est_une_date_valide(chaine):
    if len(chaine) != 10:
        return False
    if chaine[2] != "/" or chaine[5] != "/":
        return False
    for idx in range(10):
        if idx != 2 and idx != 5:
            if not est_numérique(chaine[idx]):
                return False
    return True

##########################################################################

# Ex 9
def affiche_coordonnées(x, y):
    print("Ce point a pour coordonnées (%s, %s)." % (x, y))

##########################################################################

# Ex 10
s = "abcde"
for idx in range(len(s)):
    print(idx, s[idx])
# 0 a
# 1 b
# 2 c
# 3 d
# 4 e

##########################################################################
```

```python
# Ex 11
def nb_e(s):
    compteur = 0
    for carac in s:
        if carac == "e":
            compteur +=1
    return compteur

#########################################################################

# Ex 12
def est_binaire_valide(s):
    for c in s:
        if c != "0" and c != "1":
            return False
    return True

print(est_binaire_valide("001010"))

#########################################################################

# Ex 13
# 1)
def valeur_decimale(s):
    n = len(s)
    somme = 0
    for idx in range(n):
        if s[idx] == "1":
            somme = somme + 2**(n-1-idx)
    return somme

print(valeur_decimale("00010"))

# 2)
def valeur_decimale_2(s):
    n = len(s)
    somme = 0
    for idx in range(n):
        c = s[idx]
        pt_code = ord(c)
        if pt_code >= 48 and pt_code <= 57:
            digit = int(c)
        elif 65 <= pt_code <= 70:
            digit = pt_code - 55
        somme = somme + digit*16**(n-1-idx)
    return somme

print(valeur_decimale_2("45EA"))

#########################################################################
```

```python
# Ex 14
def est_palindrome(s):
    n = len(s)
    for idx in range(n//2):
        if s[idx]!=s[n-1-idx] : # ou s[idx] != s[-idx-1]
            return False
        return True

##########################################################################

# Ex 15
def somme_chiffres(n):
    s = str(n)
    somme = 0
    for c in s:
        somme += int(c)
    return somme

def somme_tous_les_chiffres(n):
    somme = 0
    for k in range(n):
        somme += somme_chiffres(k)
    return somme

##########################################################################

# Ex 16
chaine = "abc"
s = ""
for char in chaine:
    s = s + char + s
print(s) # affiche 'abacaba'

##########################################################################

# Ex 17
def str_filter(s, mask):
    res = ""
    for carac in s:
        if carac not in mask:
            res = res + carac
    return res

##########################################################################

# Ex 18
def a_l_envers(s):
    res = ""
    for c in s:
        res = c + res
    return res
```

```python
##########################################################################

# Ex 19
def chiffre_vigenere(msg, key):
    res = ""
    n = len(key)
    idx = 0
    for c in msg:
        decalage = rang(key[idx])
        r = rang(c)
        res = res + lettre((r + decalage) % 26)
        idx = (idx + 1) % n
    return res

def dechiffre_vigenere(msg, key):
    res = ""
    n = len(key)
    idx = 0
    for c in msg:
        decalage = rang(key[idx])
        r = rang(c)
        res = res + lettre((r + 26 - decalage) % 26)
        idx = (idx + 1) % n
    return res

print(chiffre_vigenere("lyceelangevinwallon", "nsi"))
print(dechiffre_vigenere("yqkrwtnfornqaoiydwa", "nsi"))

##########################################################################

# Ex 20
reader = open ("encodage.txt", "r", encoding = "latin-1" )
s = reader.read()
reader.close()
writer = open("encodage-utf8.txt", "w", encoding="utf-8")
writer.write(s)
writer.close()
```