

```
charset="utf-8"
http-equiv="X-UA-Compatible"
Glitch Effect Dem
viewport
stylesheet
src="https://code
src="mgGlitch.901.0"
```



Exercice 1

Rédigez une fonction nommée `from_roman_numeral` renvoyant la valeur décimale de nombre romain reçu en paramètre.

```
>>> print(from_roman_numeral("V"))
5
>>> print(from_roman_numeral("XX"))
20
>>> print(from_roman_numeral("DCCC"))
800
>>> print(from_roman_numeral("MMMM"))
4000
```

Entrez un nombre : LXIX
69

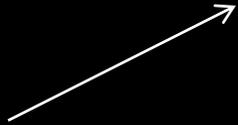
Entrez un nombre : XIV
14

Entrez un nombre : MMMDXCLXIV
3654

```
def from_roman_numeral(nombre):  
    '''La fonction "from_roman_numeral" permet de traduire un chiffre romain  
    en nombre décimal, tout en prenant en compte les différentes erreurs possibles,  
    et les soustractions.'''  
    lN=[] #création des variables  
    lS=[]  
    result=0  
    I,V,X,L,C,D,M=0,0,0,0,0,0,0
```



```
for i in nombre: ###décodage des caractères  
    if i == "I": #gestion de "I"  
        lN.append(1)  
        I+=1  
    elif i == "V": #gestion de "V"  
        lN.append(5)  
        V+=1  
    elif i == "X": #gestion de "X"  
        lN.append(10)  
        X+=1  
    elif i == "L": #gestion de "L"  
        lN.append(50)  
        L+=1  
    elif i == "C": #gestion de "C"  
        lN.append(100)  
        C+=1  
    elif i == "D": #gestion de "D"  
        lN.append(500)  
        D+=1  
    elif i == "M": #gestion de "M"  
        lN.append(1000)  
        M+=1  
    else : #vérification syntaxe  
        result="Erreur de syntaxe, recommencez."  
        break
```



Entrez un nombre : IV
4

Entrez un nombre : VX
Erreur d'empilage, recommencez.

```
while True:
    conteur=0
    for i in range(0,len(ln)-1):
        conteur+=1
        if i>=len(ln)-1:      #fin des soustractions
            break
        elif ln[i]<ln[i+1]: #gestion de soustractions
            if ln[i]==1 and (ln[i+1]==5 or ln[i+1]==10): #pour "I"
                ls.append(ln[i+1]-ln[i])
                ln.remove(ln[i])
                ln.remove(ln[i])
                continue
            elif ln[i]==10 and (ln[i+1]==50 or ln[i+1]==100): #pour "X"
                ls.append(ln[i+1]-ln[i])
                ln.remove(ln[i])
                ln.remove(ln[i])
                continue
            elif ln[i]==100 and (ln[i+1]==500 or ln[i+1]==1000): #pour "C"
                ls.append(ln[i+1]-ln[i])
                ln.remove(ln[i])
                ln.remove(ln[i])
                continue
            else :
                #vérification erreur d'empilation
                result="Erreur d'empilage, recommencez."
        else :
            continue
    break
```

```
for i in range(0, len(LN)): #liste des nombres
    result+=LN[i]
for i in range(0, len(LS)): #liste résultat soustractions
    result+=LS[i]
return result
```

```
elif i == "M":                #gestion de "M"
    lN.append(1000)
    M+=1
else :                        #vérification syntaxe
    result="Erreur de syntaxe, recommencez."
    break

                                ###vérification des nombres
if result!="Erreur de syntaxe, recommencez.":
    if I>=4 or V>=2 or X>=5 or L>=2 or C>=5 or D>=2 or M>=6:
        result="Erreur de nombre, recommencez."
                                ###gestion des soustraction des éléments
if result!="Erreur de nombre, recommencez.":
    while True:
        compteur=0
```



```
elif lN[i]==100 and (lN[i+1]==500 or lN[i+1]==1000): #pour "C"
    lS.append(lN[i+1]-lN[i])
    lN.remove(lN[i])
    lN.remove(lN[i])
    continue
else : #vérification erreur d'empilation
    result="Erreur d'empilage, recommencez."
else :
    continue
break

###calcul du résultat
if result!="Erreur d'empilage, recommencez.":
    for i in range(0,len(lN)): #liste des nombres
        result+=lN[i]
    for i in range(0,len(lS)): #liste résultat soustractions
        result+=lS[i]
    return result
else:
    return result
else:
    return result
else:
    return result
```

```
>>> print(from_roman_numeral("V"))
5
>>> print(from_roman_numeral("XX"))
20
>>> print(from_roman_numeral("DCCC"))
800
>>> print(from_roman_numeral("MMMM"))
4000
```



```
Entrez un nombre : V
5
Entrez un nombre : XX
20
Entrez un nombre : DCCC
800
Entrez un nombre : MMMM
4000
```