

Exercice 1.4.3 - Solution :

a) Définir une fonction *puissancen(...)* qui calcule et affiche la puissance n d'un nombre a

Entrees : a et n

```
def puissancen(a,n):  
    res = a**n  
    print ('{} puissance {} = {}'.format(a, n, res))
```

b) Afficher 3 au carré et 4 au cube en appelant la fonction

```
puissancen(3, 2)  
puissancen(4, 3)
```

c) Demander à l'utilisateur de **saisir a et n** , puis afficher le résultat de a puissance n

```
nb = int( input(" Entrez un nombre : ") )  
p = int( input(" Entrez la puissance : ") )  
puissancen(nb, p)
```

Exercice 1.4.3 - Solution :

d) Créer une liste avec les valeurs 1, 2, 3, ..., 10

```
liste = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
```

e) Afficher la puissance carrée des 4 premiers éléments de la liste

```
for i in range(0,4):  
    puissancen(liste[i], 2)
```

f) Afficher la puissance carrée de 3 éléments de la liste à partir du 5^{ème} élément

```
for i in range(4,4+3):  
    puissancen(liste[i], 2)
```

Exercice 1.4.3 - Solution :

g) Définir une fonction *fact(n)* qui retourne le factoriel d'un nombre n (*fact(n) = 1 x 2 x 3 ... x n*). Afficher *fact(3)*

version avec for

```
def fact(n):  
    fa=1  
    for i in range (1,n+1):  
        fa = fa*i  
    return fa
```

fact(3)

version avec récursivité

```
def factR(n):  
    if (n==1):  
        return 1  
    return n*factR(n-1)
```

factR(3)