Décomposition primaire

In [1]:

```
# Question 1

def estPremier(n):
    if n in (0,1):
        return False
    d = 2
    while d**2 <= n:
        if n%d == 0:
            return False
        d = d + 1
    return True
```

In [2]:

```
# Test
estPremier(13)
```

Out[2]:

```
True
```

In [3]:

```
estPremier(14)
```

Out[3]:

```
False
```

In [4]:

```
# Question 2
def liste_premiers(n):
    return [k for k in range(n+1) if estPremier(k)]
```

In [5]:

```
# Test
liste_premiers(17)
```

Out[5]:

```
[2, 3, 5, 7, 11, 13, 17]
```

In [8]:

```
# Question 3
def valuation_p_adique(n,p):
    k = 0
    while n%p == 0:
        n //=p # n = n//p
        k +=1 # k = k+1
    return k
```

In [9]:

```
# Test
valuation_p_adique(40,2)
```

Out[9]:

```
3
```

In [10]:

```
# Question 4
def val(n,p):
    if n%p != 0:
        return 0
    return 1 + val(n//p,p)
```

In [11]:

```
# Test
val(40,2)
```

Out[11]:

```
3
```

In [12]:

```
val(40,5)
```

Out[12]:

```
1
```

In [13]:

```
val(40,7)
```

Out[13]:

```
0
```

In [14]:
```
# Question 5
def decomposition_facteurs_premiers(n):
    return [[p,val(n,p)] for p in liste_premiers(n) if val(n,p)!=0]
```

In [15]:
```
# Test
decomposition_facteurs_premiers(40)
```

Out[15]:
```
[[2, 3], [5, 1]]
```

In [17]:
```
decomposition_facteurs_premiers(204)
```

Out[17]:
```
[[2, 2], [3, 1], [17, 1]]
```

In [ ]:

In [ ]:

In [ ]: