TP 9- Proposition de solutions

Solution 1 Division euclidienne dans $\mathbb{N}$

```
def DE(a,b):
    q=0
    while a>=b:
        a,q=a-b,q+1
    return q,a
```

Solution 2 Algorithme d'Euclide

L'algorithme d'Euclide complété :

Entrées : $a, b \in \mathbb{Z}$ avec $|b| > 0$

$B \leftarrow b$
$A \leftarrow a$
Tant que $B \neq 0$ faire
    $R \leftarrow$ reste de la division euclidienne de A par B
    $A \leftarrow B$
    $B \leftarrow R$
FinTantQue

Sortie : A le PGCD de $a$et b

```
1 def pgcd(a,b):
2     while b>0:
3         a,b=b,a%b
4     return a
```

La version détaillée est obtenu à insérant après la ligne 2, l'instruction suivante :

```
        print(a,'=',a//b,'*',b,'+',a%b)
```

Solution 3 Relation de Bezout

1. Calcul du couple de bezout et du PGCD :

```
def Bezout(a,b):
    u,v,uu,vv=1,0,0,1
    while b!=0:
        q,r=a//b,a%b
        u,v,uu,vv,a,b=uu,vv,u-q*uu,v-q*vv,b,r
    return u,v,a
```

2. Résolution du problème chinois

```
def chinois(a,b,p,q):
    u,v,d= Bezout(p,q)
    return (a*v*q+b*u*p)%(p*q), p*q
```

Solution 4 Décomposition en facteurs premiers

```
def valuation(p,n):
    assert n>0,'ENC'
    a=0
    while n%p==0:
        n,a=n//p,a+1
    return a
```

```
def dfp(n):
    assert n>1,'ENC'
    L=[]
    p=2
    while p*p<=n:
        a=valuation(p,n)
        if a>0:
            L.append([p,a])
            n=n//p**a
        p=p+1
    if n>1:
        L.append([n,1])
    return L
```

Solution 5 Méthode du crible d'Eratosthène

```
def crible(n):
    L=list(range(n+1))
    L[1]=0
    p=2
    while p*p<=n:
        if (L[p]!=0):
            L[p*p::p]=[0]*(n//p-p+1)
        p=p+1
    P=[p for p in L if p!=0]
    return P
```

ou

```
def crible2(n):
    L=list(range(2,n+1))
    i=0
    while i<len(L) and L[i]**2<=n:
        j=i+1
        while j<len(L):
            if L[j]%L[i]==0:
                L.pop(j)
            else:
                j=j+1
        i=i+1
    return L
```


Lycée Saint-Exupery – maffre.saintexupery@gmail.com – MPSI