

**TP échec : correction**

1. 

```
def Indice(i,j) :
    return 8*i + j
```

2. 

```
def Coord(n) :
    return [n//8,n \%8] # antislash inutile en Python
```

3. CasA(0)= [17,10] et CasA(39)= [45,54,22,29].  
Plus généralement, CasA(n) est la liste des indices des cases accessibles pour un cavalier placé en case d'indice n.

4. 

```
def Init() :
    global ListeCA, ListeCoups
    ListeCoups=[]
    ListeCA=[CasA(n) for n in range(64)]
```

5. Cela renvoie [17,10].

6. 1. 

```
def OccupePosition(n) :
    global ListeCA, ListeCoups
    ListeCoups.append(n)
    critique = False
    for k in ListeCA[n] :
        ListeCA[k].remove(n)
        if ListeCA[k]==[] :
            critique = True
    return critique
```

2. 

```
def LiberePosition() :
    global ListeCA, ListeCoups
    n=ListeCoups.pop()
    for k in ListeCA[n] :
        ListeCA[k].append(n)
```

3. 

```
def TestePosition(n) :
    global ListeCA, ListeCoups
    if OccupePosition(n) :
        if len(ListeCoups)==63 :
            return True
        else :
            LiberePosition()
            return False
    else :
        L=ListeCA[n].copy()
        for k in L :
            if TestePosition(k) :
                return True
        LiberePosition()
        return False
```

7. 1. 

```
def valuation(n) :
    global ListeCA
    return len(ListeCA[n])
```

2. Le tri fusion est un grand classique qu'il faut avoir étudié au moins une fois dans sa vie! L'algorithme TriFusion est toujours récursif mais l'algorithme Fusion peut être itératif ou récursif. On propose ici une version itérative.

```
def Fusion(A,B) :
    L=[]
    i=0
    j=0
    while i<len(A) and j<len(B) :
        if valuation(A[i])<valuation(B[j]) :
            L.append(A[i])
            i=i+1
        else :
            L.append(A[j])
            j=j+1
    if i<len(A) :
        L=L+A[i:]
    else :
        L=L+B[j:]
    return L
```

```
3. def TriFusion(L) :  
    if len(L)<=1 :  
        return L  
    else :  
        n=len(L)//2  
        return Fusion(TriFusion(L[:n]),TriFusion(L[n:]))
```

```
4. def TestePosition2(n) :  
    global ListeCA, ListeCoups  
    if OccupePosition(n) :  
        if len(ListeCoups)==63 :  
            return True  
        else :  
            LiberePosition()  
            return False  
    else :  
        L=TriFusion(ListeCA[n])  
        for k in L :  
            if TestePosition(k) :  
                return True  
        LiberePosition()  
        return False
```

Exemple : TestePosition2(1)=True et on a alors

ListeCoups= [1, 16, 33, 48, 58, 52, 62, 47, 53, 59, 44, 50, 56, 41,  
51, 57, 42, 36, 30, 45, 60, 54, 39, 22, 7, 13, 23, 29, 35, 25, 40, 34, 24,  
9, 26, 32, 49, 43, 37, 31, 14, 4, 19, 2, 8, 18, 3, 20, 5, 15, 21, 6, 12, 27,  
10, 0, 17, 11, 28, 38, 55, 61, 46].