

Corrigé du TP Informatique 8

Exercice 1

1. On saisit :

```
def detect(elt,L):  
    for x in L:  
        if x==elt:  
            return True  
    return False
```

2. On saisit :

```
def pos(elt,L):  
    res=[]  
    for k in range(len(L)):  
        if L[k]==elt:  
            res.append(k)  
    return res
```

Exercice 2

On saisit :

```
def elt(L):  
    res=[]  
    for x in L:  
        if not detect(x,res):  
            res.append(x)  
    return res
```

Exercice 3

1. On saisit :

```
def mini(L):  
    res=L[0]  
    for x in L:  
        if res>x:  
            res=x  
    return res
```

2. On saisit :

```
def mini_pos(L):
    n=len(L)
    res,ind=L[0],0
    for k in range(n):
        if res>L[k]:
            res,ind=L[k],k
    return res,ind
```

3. On saisit :

```
def mini_last_pos(L):
    n=len(L)
    res,ind=L[0],0
    for k in range(n):
        if res>=L[k]:
            res,ind=L[k],k
    return res,ind
```

Exercice 4

1. On saisit :

```
def maxi_mini(L):
    """Maximum et minimum de L"""
    m,M=L[0],L[0]
    for x in L:
        if M<x:
            M=x
        if m>x:
            m=x
    return m,M
```

2. On saisit :

```
def maxi_mini_pos(L):
    m,ind_m=L[0],0
    M,ind_M=L[0],0
    n=len(L)
    for k in range(n):
        if M<L[k]:
            M,ind_M=L[k],k
        if m>L[k]:
            m,ind_m=L[k],k
    return [[m,ind_m],[M,ind_M]]
```

Exercice 5

1. On saisit :

```
def max2(L):
    """Détermine 2 plus grands éléments de L"""
    a,b=L[0],L[1]    # a sera max(liste) et b=max(liste\{a})
    if a<b:
        a,b=b,a      # on réordonne éventuellement a et b
    for k in range(2,len(L)):
        c=L[k]
        if c>a:       # si terme courant > a
            a,b=c,a   # a prend sa valeur et b celle de a
        elif c>b:     # si terme courant <=a et >b
            b=c        # b prend sa valeur (a inchangé)
    return a,b
```

2. Il suffit de garder la mémoire de la position d'un élément lors de son affectation. On saisit :

```
def max2_ind(L):
    """Détermine 2 plus grands éléments avec index de L"""
    a=L[0],0
    b=L[1],1
    if a[0]<b[0]:
        a,b=b,a
    for k in range(2,len(L)):
        c=L[k],k
        if c[0]>a[0]:
            a,b=c,a
        elif c[0]>b[0]:
            b=c
    return a,b
```

Exercice 6

1. On saisit :

```
def maxi_occ1(L):
    """Détermination de toutes les occurrence du maximum de L"""
    M=L[0]
    for x in L:      # recherche du maximum
        if M<x:
            M=x
    res,n=[],len(L)
    for k in range(n):    # parcourt des indices de L
        if L[k]==M:       # si terme courant=M
            res.append(k)  # on mémorise sa position
    return res
```

2. On saisit :

```

def maxi_occ2(L):
    """Détermination de toutes les occurrence du maximum de L"""
    res=[0]
    M=L[0]
    for k in range(1,len(L)):
        if M<L[k]:          # si terme courant plus grand que M
            res=[k]         # on oublie positions anciennes
            M=L[k]          # on mémorise le terme courant et sa position
        elif M==L[k]:      # si doublon de M
            res.append(k)    # on mémorise sa position
    return res

```

Exercice 7

On saisit :

```

def preder(L,elt):
    n=len(L)
    g,d=None,None
    for k in range(n):
        if L[k]==elt:
            d=k
        if L[n-(k+1)]==elt:
            g=n-(k+1)
    return g,d

```