TP08 : boucles imbriquées

Corrigé

Lycée Chateaubriand – PCSI1

1 Matrices (ou listes de listes)

1. Il suffit de renvoyer la liste d'indice `k` de `m`.

```
def ligne(m, k):
    assert k < len(m)
    return m[k]
```

2. Les éléments de la colonne d'indice `k` sont tous les `mat[i][k]` pour `i` allant de 0 à `len(mat)-1`.

```
def colonne(m, k):
    assert k < len(m[0])
    c = []
    for i in range(len(m)):
        c.append(m[i][k])
    return c
```

2 Dessiner dans une matrice

4.

```
def diag1(n):
    m=[ [0]*n for _ in range(n) ]
    for i in range(n):
        m[i][i]=1
    return m

def diag2(n):
    m=[ [0]*n for _ in range(n) ]
    for i in range(n):
        m[n-1-i][i]=1
    return m
```

5.

```
def croix(n):
    assert(n%2==1)
    m=[ [0]*n for _ in range(n) ]
    for i in range(n):
        m[n//2][i]=1
        m[i][n//2]=1
    return m
```

6. On utilise des définitions en compréhension, qui permettent de traiter toutes ces questions de façon simple et uniforme.

```
def triangle1(n):
    m=[ [0]*n for _ in range(n) ]
    for i in range(n):
        for j in range(i+1):
            m[i][j]=1
    return m

```

```
def triangle2(n):
    m=[ [0]*n for _ in range(n) ]
    for i in range(n):
        for j in range(n-i-1,n):
            m[i][j]=1
    return m
```

7.
```
def triangle3(n):
    m=[ [0]*n for _ in range(n) ]
    for i in range(n):
        for j in range(n-i):
            m[i][j]=1
    return m

def triangle4(n):
    m=[ [0]*n for _ in range(n) ]
    for i in range(n):
        for j in range(i,n):
            m[i][j]=1
    return m
```

3 Sommes doubles

8. On imbrique deux boucles `for`, et on additionne chaque valeur dans une variable `res`.
```
def somme(m):
    s = 0
    for ligne in m:
        for nb in ligne:
            s = s+nb
    return s
```

9. On définit la matrice en extension (en faisant attention à bien choisir les bornes de la fonction `range`), et on appelle la fonction précédente.
```
def somme_double(n):
    mat = [ [1/(i**2+j**2) for j in range(1,n+1)] for i in range(1,n+1) ]
    return somme(mat)
```

10. Pas de grande difficulté, mais il faut faire attention à bien faire correspondre les indices.

4 Deux éléments les plus proches dans une liste

11.
```
def plusProches(l):
    dist=abs(l[1]-l[0])
    iopt=0
    jopt=1
    for i in range(len(l)):
        for j in range(len(l)):
            if j != i and abs(l[i]-l[j])<dist:
                dist=abs(l[i]-l[j])
                iopt=i
                jopt=j
    return iopt, jopt
```