1. On a par exemple :

```
>>>L = [2, 4, 6]
>>>L[-1]
6
>>>L[len(L) - 1]
6
```

2. (a) Avec une boucle *for* :

```
L = []
for i in range(1, 1000):
    if i % 2 == 1: # test de l'imparité de i
        L.append(i)
```

   (b) Avec une boucle *while* :

```
M = []
i = 1
while i < 1000:
    M.append(i)
    i = i + 2
```

   (c) Avec une liste en compréhension :

```
N = [ 2 * i + 1 for i in range(500)]
```

3. Lors du premier passage dans la boucle la commande *return* stoppe l'exécution de la fonction et la valeur 0 est renvoyée.

4. La liste L vaut [0, 1, 2]. Pour chaque valeur de i entre 0 et 2 toutes les valeurs de j entre 0 et 2 sont parcourues, ce qui donne :

```
00
01
02
10
11
12
20
21
22
```

5. Voici les résultats, il faut juste suivre pas à pas les différentes conditions selon les valeurs de n :

```
>>>[test2(k) for k in range(10)]
[0, 0, 2, 3, 4, 3, 3, 6, 7, 8]
```

6. Voici une fonction réalisant ceci. Si la liste est vide, on ne rentre pas dans la boucle et le programme renvoie 1, ce qui est cohérent car le produit vide vaut 1.

```
def mul(L):
    """renvoie le produit des éléments de la liste L"""
    p = 1
    for i in range(len(L)):
        p = p * L[i]
    return(p)
```

7. Cela renvoie une erreur car a est une variable locale qui n'existe qu'au sein de la fonction tandis que l'affichage se fait en dehors de la fonction :

```
NameError: name 'a' is not defined
```

8. Voici une fonction qui réalise ceci :

```
import random as rd

def mel(L):
    """melange la liste L"""
    M = [] #la nouvelle liste
    while L != []:
        i = rd.randint(0, len(L) - 1) #choix d'un indice au hasard
        M.append(L[i]) #on ajoute l'élément à M
        L.pop(i) # on le supprime dans L
    return(M)
```

9. Voici une fonction à l'aide d'un boucle for :

```
def rev(ch):
    """écrit la chaine de caractères dans l'autre sens"""
    ch2 = "" # la nouvelle chaine de caractère, vide initialement
    for i in range(0, len(ch)):
        ch2 = ch2 + ch[len(ch) - i - 1] # on ajoute les éléments en partant de la fin
    return(ch2)
```