Exercice 1.
1)

```python
def suite(n):
    u = 1
    L = [u]
    for k in range(n):
        u = -u**2 + 1
        L.append(u)
     return L
```

2) On appelle `suite(100)` depuis la console en affectant le résultat renvoyé à une liste dont on extrait les listes des termes de rangs pairs et impairs par extraction de liste avec pas 2 et premier indice respectivement 0 et 1.

```
In[1] : L = suite(100)

In[2] : L[0::2]
Out[2]: [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]

In[3] : L[1::2]
Out[3]:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
```

3) Conjecture : la suite extraite des termes de rangs pairs est stationnaire en 1 ; celle des termes de rangs impairs est stationnaire en 0. Par récurrence avec pour propriété de récurrence : $\mathscr{P}(k)$: "$u_{2k} = 1$, $u_{2k+1} = 0$". (I) Pour $k = 0$: $u_0 = 1$ et $u_1 = -u_0^2 + 1 = -1 + 1 = 0$; $\mathscr{P}(0)$ est vérifiée.

(H) Supposons $\mathscr{P}(k)$ vraie pour $k \in \mathbb{N}$ fixé : $u_{2k} = 1$ et $u_{2k+1} = 0$.
D'après la relation de récurrence appliquée deux fois :

$$u_{2k+2} = -u_{2k+1}^2 + 1 = 1 \quad ; \quad u_{2k+3} = -u_{2k+2}^2 + 1 = -1 + 1 = 0$$

Ainsi $u_{2(k+1)} = 1$ et $u_{2(k+1)+1} = 0$: $\mathscr{P}(k+1)$ est vérifiée.
Ceci conclut la récurrence et prouve la conjecture.

Exercice 2.

```python
def fibonacci(n):
    u, v = 0, 1
    L = [u,v]
    for k in range(n-1):
        u, v = v, u+v
        L.append(v)
    return L
```

Exercice 3.

```python
def somme(L):
    S = 0
    for x in L:
        S = S+x
     return S
```

Exercice 4. Obtention de la ligne n du triangle de Pascal :

```python
## 1)
def coefBinomiaux(n):
    L = [1]
    for m in range(1,n+1):
        L2 = [1]
        for k in range(1,m):
            L2.append(L[k-1]+L[k])
        L2.append(1)
        L = L2
    return L
```

Exemple :

```
>>> coefBinomiaux(3)
[1, 3, 3, 1]

>>> coefBinomiaux(5)
[1, 5, 10, 10, 5, 1]
```

Affichage de la formule du binôme :

```python
## 2)

def binome(n):
    L = coefBinomiaux(n)
    ch = "a^" + str(n)
    for k in range(1,n):
        ch = ch + "+" + str(L[k]) + "a^" \
                        + str(n-k) + "b^" + str(k)
    ch = ch + "+b^" + str(n)
    print(ch)
```

Exemple :

```
>>> binome(3)
a^3+3a^2b^1+3a^1b^2+b^3

>>> binome(5)
a^5+5a^4b^1+10a^3b^2+10a^2b^3+5a^1b^4+b^5
```