```python
import numpy as np
import matplotlib.pyplot as plt


## Exercice 1

def f(x):
    return np.floor(np.log(x))

def images(n):
    return np.array([f(k) for k in range(1,n+1)])
    # ou: return f(np.arange(1,n+1))

def imagesbis(n):
    k = np.arange(1,n+1)
    return f(k) / k
    # ou : return np.array([f(k)/k for k in range(1,n+1)])


## Exercice 2

plt.title("la fonction cos")
X = np.linspace(0, 2*np.pi)
plt.plot(X,np.cos(X),'r:s')
plt.show()


## Exercice 3

plt.figure()
plt.title("la fonction exp")
X = np.linspace(-2, 2)
plt.plot(X,np.exp(X),'r')
plt.show()


plt.figure()
plt.title("la fonction ln")
X = np.linspace(0.3, 4)
plt.plot(X,np.log(X),'b')
plt.show()


plt.figure()
plt.title("la fonction racine carrée")
X = np.linspace(0, 5)
plt.plot(X,np.sqrt(X),'g')
plt.show()


def f(x):
    return x**3 + x - 2

plt.figure()
X = np.linspace(-3, 3)
Y = np.array([f(x) for x in X])
plt.plot(X,Y,'k')
plt.show()
```

```
## Exercice 4

X1 = np.linspace(0,5)
X2 = np.linspace(0,2.2)
Y2 = np.array([x**2 for x in X2])
X3 = np.linspace(0,4)

plt.plot(X1, np.sqrt(X1), 'g-', label="racine carrée")
plt.plot(X2, Y2, 'r-', label="x->x^2")
plt.plot(X3, X3, 'k--')
plt.legend(loc = "upper right")
ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))
ax.spines['bottom'].set_position(('data',0))
plt.xticks([1, 2, 3, 4, 5])
plt.yticks([1, 2, 3, 4, 5])
plt.axis([-1.2, 6, -0.2, 5.5])

plt.show()



## Exercice 5

X = np.linspace(0,5)

plt.subplot(2,2,1)
plt.plot(X,np.cos(4*X)*np.exp(X),'r-')
plt.plot(X,np.exp(X),'k--')
plt.plot(X,-np.exp(X),'k--')
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['right'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.spines['bottom'].set_position(('data',0))
plt.title("f(x)=exp(x) cos(4x)")

plt.subplot(2,2,2)
plt.plot(X,np.cos(4*X)*np.exp(-X),'r-')
plt.plot(X,np.exp(-X),'k--')
plt.plot(X,-np.exp(-X),'k--')
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['right'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.spines['bottom'].set_position(('data',0))
plt.title("f(x)=exp(-x) cos(4x)")

plt.subplot(2,2,3)
plt.plot(X,np.exp(-X),'r-')
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['right'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.spines['bottom'].set_position(('data',0))
plt.title("f(x)=exp(-x)")
plt.ylim(-1,1)
```

```
plt.subplot(2,2,4)
plt.plot(X,X*np.exp(-X),'r-')
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['right'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.spines['bottom'].set_position(('data',0))
plt.title("f(x)=x.exp(-x)")
plt.ylim(-1,1)

plt.show()
```