In [16]:

```
%matplotlib qt
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import os
```

In [2]:

```
SHAPE = (64, 64)
kls = [10, 20, 30, 40, 50, 100]
ITER = 100
size_x = SHAPE[0]*SHAPE[1]
INSTANCES = 165
img_idx = [0, 20, 40, 60, 80]
```

In [3]:

```
img = np.load('/home/bhavy/Dropbox/7th-semester/courses/ML/Assignments/Assignment/h
```

In [4]:

```
data = img['X']
```

In [12]:

```
fig = plt.figure(1)
fig.suptitle("Original Images", fontsize=16)
for i, index in enumerate(img_idx):
    plt.subplot(5, 1, i+1)
    imgplot = plt.imshow(data[index].reshape(SHAPE),cmap = 'gray' )
plt.savefig('original/' +'img-'+ str(index)+'.jpg')
```

In [6]:

```
mean = data.mean(axis = 0)
data_zero_centered  = data - mean
```

In [7]:

```
def compute_z(W, a):
    M = np.dot(np.transpose(W), W)
    return np.dot(np.linalg.inv(M), np.dot(np.transpose(W) , a) )
```

In [8]:

```
def compute_W(Z, data_zero_centered):
    a = np.zeros((size_x, size_z))
    b = np.zeros((size_z, size_z))
    for i in range(INSTANCES):
        a = a +np.dot(data_zero_centered[i].reshape((size_x, 1)) ,np.transpose(Z[i]
        b = b + np.dot( Z[i].reshape((size_z, 1))  , np.transpose(Z[i].reshape((siz
    return np.dot(a, np.linalg.inv(b))
```

In [21]:

```
for k in kls:
    size_z = k
    W = np.random.normal(size=(size_x, size_z))
    Z = np.zeros((165, size_z))
    for itera in range(ITER):
        for i in range(165):
            Z[i] = compute_z(W, data_zero_centered[i])
        W = compute_W(Z, data_zero_centered)

    #plotting images
    fig = plt.figure()
    fig.suptitle("Reconstructed Images for K = " + str(k) , fontsize=16)
    for idx, i in enumerate(img_idx):
        plt.subplot(5,1, idx + 1)
        imgplot = plt.imshow((mean  +  np.dot(W, Z[i])).reshape(SHAPE),cmap = 'gray
    plt.savefig(str(k) + '/' + 'reconstructed-' + str(k) + '.jpg')
    plt.close()
    #plotting basis
    fig = plt.figure()
    fig.suptitle("Basis Images for K = " + str(k) , fontsize=16)
    for basis in range(10):
        plt.subplot(5,2, basis + 1)
        imgplot = plt.imshow(( W[:, basis]).reshape(SHAPE),cmap = 'gray' )
    plt.savefig(str(k) + '/' + 'basis-' + str(k)  +'.jpg')
    plt.close()
```