Ex. 1 – Features Quantization

```
train_des_labels = []
test_des_labels = []

# Compute pairwise distance matrix
train_distances = sklearn.metrics.pairwise_distances(train_des,
visual_words)
# Find the closest visual word to each training descriptor
train_des_labels = np.argmin(train_distances, axis=1)

# Compute pairwise distance matrix
test_distances = sklearn.metrics.pairwise_distances(test_des, visual_words)
# Find the closest visual word to each testing descriptor
test_des_labels = np.argmin(test_distances, axis=1)

print(f"train_des_labels shape: {train_des_labels.shape}")
print(f"test_des_labels shape: {test_des_labels.shape}")
```

Ex. 2 – Histograms of visual words

```
bovw_train = []

# Loop over training labels
for label in range(len(train_images)):
  # Loop over images
  bovw_train_label = []
  for idx_image in range(len(train_images[label])):
    # Select descriptors associated to the image
    mask = np.logical_and(label_mask_train == label, image_mask_train ==
idx_image)
    # Retrieve labels of visual words associated to the selected descriptors
    sel_des_labels = train_des_labels[mask]

    # Create histogram of visual words
    h = np.histogram(sel_des_labels, bins=n_visual_words)[0].reshape(1, -1)

    # Save current bovw
    bovw_train_label.append(h)

  # Save list of bovw(s)
  bovw_train.append(bovw_train_label)

print(f"len(bovw_train): {len(bovw_train)}")
for label in range(len(bovw_train)):
  print(f"len(bovw_train[{label}]): {len(bovw_train[label])}")

print(f"\nbovw_train[0][0]:\n{bovw_train[0][0]}")
```

```
bovw_test = []

# Loop over training labels
for label in range(len(test_images)):
  # Loop over images
  bovw_test_label = []
  for idx_image in range(len(test_images[label])):
    # Select descriptors associated to the image
    mask = np.logical_and(label_mask_test == label, image_mask_test ==
idx_image)
    # Retrieve labels of visual words associated to the selected descriptors
    sel_des_labels = test_des_labels[mask]
    # Create histogram of visual words
    h = np.histogram(sel_des_labels, bins=n_visual_words)[0].reshape(1, -1)

    # Save current bovw
    bovw_test_label.append(h)

  # Save list of bovw(s)
  bovw_test.append(bovw_test_label)

print(f"len(bovw_test): {len(bovw_test)}")
for label in range(len(bovw_test)):
  print(f"len(bovw_test[{label}]): {len(bovw_test[label])}")

print(f"\nbovw_test[0][0]:\n{bovw_test[0][0]}")
```

Ex. 3 – (NN classifier)

```
# Compute pairwise distances
dist_matrix_l2 = sklearn.metrics.pairwise_distances(bovw_test_mat,
bovw_train_mat)
# Compute closest train histogram
closest_bovw = np.argmin(dist_matrix_l2, axis=1)
# Retrieve corresponding label
predictions = train_labels[closest_bovw]
```

Ex. 3.1 – NN classifier (Chi2)

```
# Compute pairwise distances
dist_matrix_chi2 = sklearn.metrics.pairwise_distances(bovw_test_mat,
bovw_train_mat, metric=chi2_distance)
# Compute closest train histogram
closest_bovw = np.argmin(dist_matrix_chi2, axis=1)
# Retrieve corresponding label
predictions = train_labels[closest_bovw]
```