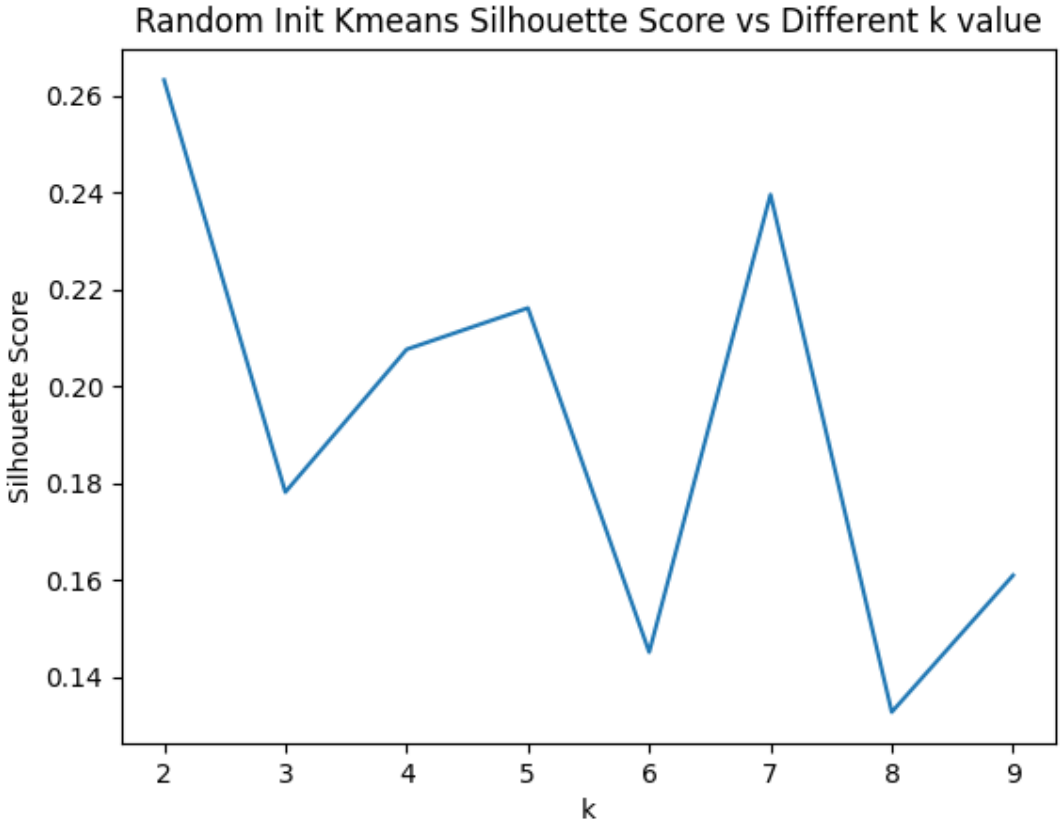
2) Apply your implementation of KMeans to cluster the mouse heart tissue dataset. Use the provided PCA implementation to reduce the dimensionality to 100 first. Use random initialization to produce clusterings for k from 2 to 9 and plot the silhouette coefficient for all clusterings in your report file. What is the best k? [25 marks]

Code cell :

```
    ran_scores = []
    for k in range(2, 10):
        print("value of k",k)
        ran_km = KMeans(n_clusters=k, init='random')
        labels = ran_km.fit(X)
        score = ran_km.silhouette(labels, X)
        ran_scores.append(score)

    fig, ax = plt.subplots()
    ax.plot(np.arange(2, 10), ran_scores)
    ax.set_xlabel('k')
    ax.set_ylabel('Silhouette Score')
    ax.set_title('Random Init Kmeans Silhouette Score vs Different k
value')
    plt.savefig('Task2.png')
    print(ran_scores)
```

Value of K	Silhouette Score
2	0.2631522538922816
3	0.1781238074426612
4	0.20757633210389315
5	0.21610370785210112
6	0.14511482182050933
7	0.23951223452111986
8	0.1327560169890546
9	0.16096655202582336



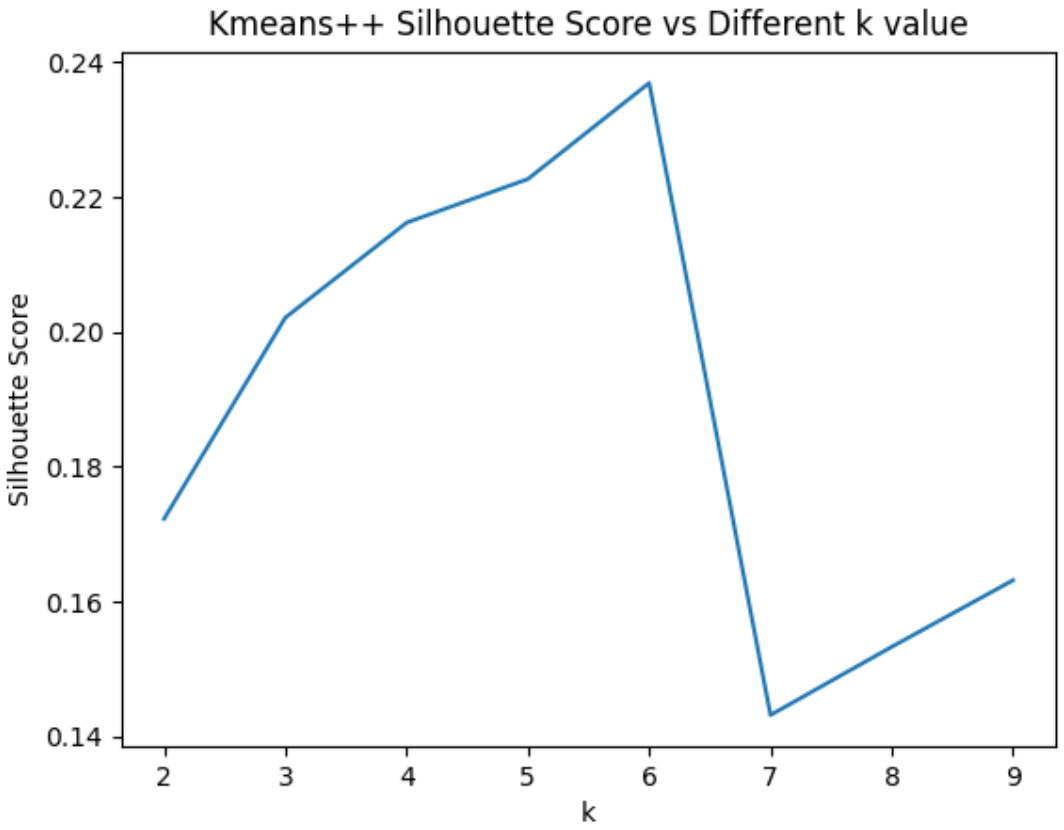
We see that the highest value of silhouette score exists for k = 2. Therefore we conclude that the optimal number of clusters for the given data is 2.

3) Produce clusterings for k from 2 to 9 using KMeans++ initialization. Report the silhouette coefficients again. What is the best k now? What is your conclusion? [25 Marks]

```
kmpp_scores = []
    for k in range(2, 10):
        kmpp = KMeans(n_clusters=k, init='kmeans++')
        labels = kmpp.fit(X)
        score = kmpp.silhouette(labels, X)
        kmpp_scores.append(score)

    fig, ax = plt.subplots()
    ax.plot(np.arange(2, 10), kmpp_scores)
    ax.set_xlabel('k')
    ax.set_ylabel('Silhouette Score')
    ax.set_title('Kmeans++ Silhouette Score vs Different k value')
    plt.savefig('Task3.png')
```

Value of K	Silhouette Score
2	0.1722928889598305
3	0.20211939274532664
4	0.21619892543355484
5	0.22265895754993506
6	0.23689835788682956
7	0.1432096493362324
8	0.15332520690836368
9	0.16319542109236862



We see that the highest value of silhouette score when using Kmeans++ exists for k = 6. Therefore we conclude that the optimal number of clusters for the given data is 6.

4) Use a scatter plot to visualize the clusters with the best k from task 2. You should reduce the dimensionality further to 2 in order to be able to visualize the clusters in a 2D plot (Note that this is only for visualization and you still need to perform KMeans using 100 features). Color-code the cells based on their clusters. Include this plot in your report file.

Best K from Task 2 is 2 (score of 0.2631522538922816)

```
ran_scores = []
for k in range(2, 10):
    print("value of k",k)
    ran_km = KMeans(n_clusters=k, init='random')
    labels = ran_km.fit(X)
    score = ran_km.silhouette(labels, X)
    ran_scores.append(score)

fig, ax = plt.subplots()
ax.plot(np.arange(2, 10), ran_scores)
ax.set_xlabel('k')
ax.set_ylabel('Silhouette Score')
```

```
   ax.set_title('Random Init Kmeans Silhouette Score vs Different k
value')
   plt.savefig('Task2.png')



  n_classifiers = np.argmax(ran_scores) + 2
   X_2 = PCA(heart.X, 2)
   model = KMeans(n_clusters=n_classifiers, init='random')
   labels = model.fit(X_2)
   visualize_cluster(X_2, labels, labels)
```