3. Use sklearn. datasets import load_iris use k-neighbour classifier to classify the three flowers to setosa, vesicolor and Virginica.

In [9]:

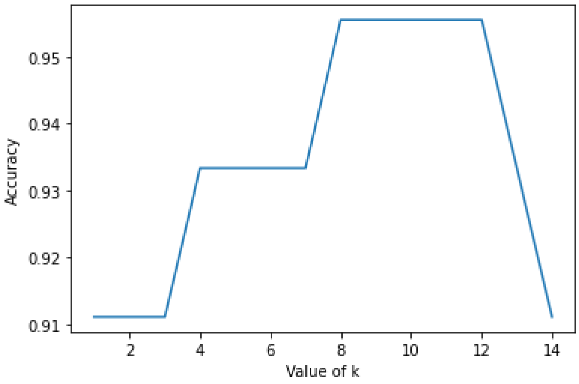
```python
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
import matplotlib.pyplot as plt
iris = load_iris()
print(iris.target_names)
print(iris.data.shape)
x = iris.data[:, :4]
y = iris.target
X_train,X_test,Y_train,Y_test = train_test_split(x,y,test_size=0.3)
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
print(X_train.shape)
print(X_test.shape)
range_k = range(1,15)
scores = {}
scores_list = []
for k in range_k:
    classifier = KNeighborsClassifier(n_neighbors=k)
    classifier.fit(X_train,Y_train)
    Y_pred = classifier.predict(X_test)
    scores[k] = metrics.accuracy_score(Y_test,Y_pred)
    scores_list.append(metrics.accuracy_score(Y_test,Y_pred))
result = metrics.confusion_matrix(Y_test,Y_pred)
print("Confusion Matrix:\n",result)
result1 = metrics.classification_report(Y_test,Y_pred)
print("Classification Report:\n",result1)
plt.plot(range_k,scores_list)
plt.xlabel("Value of k")
plt.ylabel("Accuracy")
classifier = KNeighborsClassifier(n_neighbors=8)
classifier.fit(X_train,Y_train)
classes = {0:'setosa', 1:'versicolor', 2:'virginicia'}
x_new = [[1,1,1,1],[4,3,1.3,0.2]]
y_predict = classifier.predict(x_new)
print(classes[y_predict[0]])
print(classes[y_predict[1]])
```

```
['setosa' 'versicolor' 'virginica']
(150, 4)
(105, 4)
(45, 4)
Confusion Matrix:
 [[14  0  0]
 [ 0 13  1]
 [ 0  3 14]]
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        14
           1       0.81      0.93      0.87        14
           2       0.93      0.82      0.87        17

    accuracy                           0.91        45
   macro avg       0.92      0.92      0.91        45
weighted avg       0.92      0.91      0.91        45

virginicia
virginicia
```



In [ ]: