In [28]:
```
import sklearn
from sklearn.utils import shuffle
from sklearn.neighbors import KNeighborsClassifier
from sklearn import linear_model, preprocessing
import pandas as pd
import numpy as np
```

In [29]:
```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving knn_data_sample.csv to knn_data_sample (1).csv
```

In [30]:
```
data = pd.read_csv("knn_data_sample.csv")
data
```

Out[30]:

	x1	y1	z1	x2	y2	z2	FallOrNot
0	1	2	3	2	1	3	-
1	2	1	3	3	1	2	-
2	1	1	2	3	2	2	-
3	2	2	3	3	2	1	-
4	6	5	7	5	6	7	+
5	5	6	6	6	5	7	+
6	5	6	7	5	7	6	+
7	7	6	7	6	5	6	+

In [31]:
```
x1 = list(data["x1"])
y1 = list(data["y1"])
z1 = list(data["z1"])
x2 = list(data["x2"])
y2 = list(data["y2"])
z2 = list(data["z2"])
fallOrNot = list(data["FallOrNot"])
```

In [32]:
```
X = list(zip(x1, y1,z1, x2, y2, z2))
Y = list(fallOrNot)
```

In [33]:
```
x_train, x_test, y_train, y_test = sklearn.model_selection.train_test_split(X,
Y, test_size=0.1)
```

In [34]:
```
model = KNeighborsClassifier(n_neighbors=5)
```

In [35]:
```
model.fit(x_train, y_train)
```

Out[35]:
```
KNeighborsClassifier()
```

In [36]:
```
model.score(x_test, y_test)
```

Out[36]:
```
1.0
```

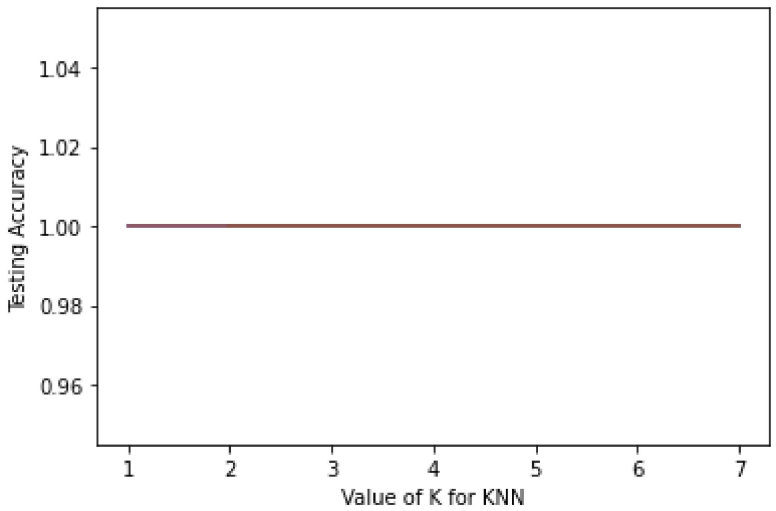
In [37]:
```
print(model.predict([(7, 6, 5, 5, 6, 7)]))
```

```
['+']
```

In [38]:
```
from sklearn import metrics
import matplotlib.pyplot as plt

# allow plots to appear within the notebook
%matplotlib inline

scores = []

# We use a loop through the range 1 to 26
# We append the scores in the dictionary
for k in x_train:
    y_pred = model.predict(x_test)
    scores.append(metrics.accuracy_score(y_test, y_pred))

print(scores)

# plot the relationship between K and testing accuracy
# plt.plot(x_axis, y_axis)
plt.plot(x_train, scores)
plt.xlabel('Value of K for KNN')
plt.ylabel('Testing Accuracy')
```

```
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
```

Out[38]:
```
Text(0, 0.5, 'Testing Accuracy')
```



In [38]: