Q. no. 2) Alter the n_neighbors value and see the difference in the output and also change the type of distance by altering the p value of the distance metric

Using k ranging from 1 to 10 and distance p = 1 (Manhattan)

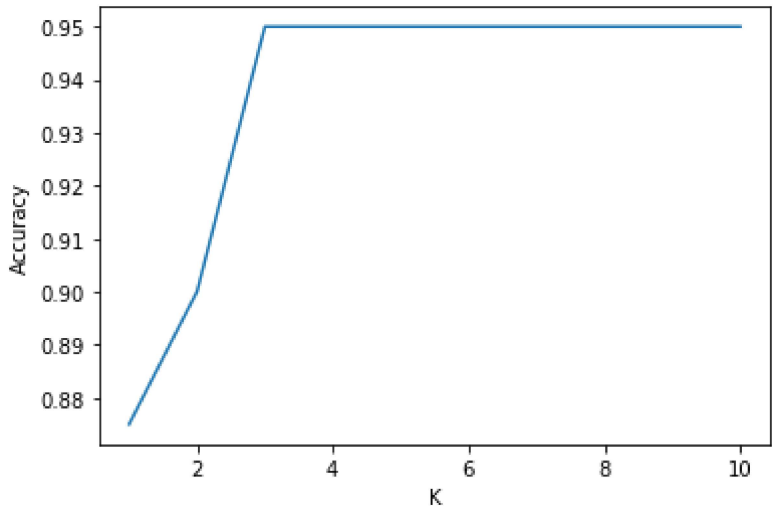
In [20]:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
acc = []
k_rng = range(1,11)
for k in k_rng:
    clf = KNeighborsClassifier(n_neighbors = k, p =1)
    clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    acc.append(metrics.accuracy_score(y_test, y_pred))
```

In [21]:

```
plt.xlabel('K')
plt.ylabel('Accuracy')
plt.plot(k_rng,acc)
print(acc)
```

```
[0.875, 0.9, 0.95, 0.95, 0.95, 0.95, 0.95, 0.95, 0.95, 0.95]
```



Using k ranging from 1 to 10 and distance p = 2 (Euclidean)

In [22]:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
acc1 = []
k_rng1 = range(1,11)
for k in k_rng1:
    clf1 = KNeighborsClassifier(n_neighbors = k, p =2)
    clf1.fit(X_train,y_train)
    y_pred1 = clf1.predict(X_test)
    acc1.append(metrics.accuracy_score(y_test, y_pred1))
```

In [23]:

```
plt.xlabel('K')
plt.ylabel('Accuracy')
plt.plot(k_rng1,acc1)
print(acc1)
```

```
[0.8625, 0.8875, 0.95, 0.95, 0.95, 0.95, 0.95, 0.95, 0.95, 0.95]
```