In [1]:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score,roc_auc_score,confusion_matrix,classification_report
iris=load_iris()
x=iris.data
y=iris.target
```

In [2]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
log_reg=LogisticRegression(max_iter=1000)
```

In [3]:

```
log_reg.fit(x_train,y_train)
y_pred=log_reg.predict(x_test)
accuracy=round(accuracy_score(y_test,y_pred)*100,2)
print("Accuracy:",accuracy)
```

```
Accuracy: 100.0
```

In [4]:

```
precision=precision_score(y_test,y_pred,average="weighted")
print("Precision:",precision)
```

```
Precision: 1.0
```

In [5]:

```
recall=recall_score(y_test,y_pred,average="weighted")
print("Recall:",recall)
```

```
Recall: 1.0
```

In [6]:

```
f1=f1_score(y_test,y_pred,average="weighted")
print("f1score:",f1)
```

```
f1score: 1.0
```

In [7]:

```
roc_auc=roc_auc_score(y_test,log_reg.predict_proba(x_test),multi_class='ovr')
print("ROC AUC score:",roc_auc)
```

```
ROC AUC score: 1.0
```

In [8]:

```
conf_matrix=confusion_matrix(y_test,y_pred)
print("Confusion matrix:",conf_matrix)
```

```
Confusion matrix: [[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

In [9]:

```
result1=classification_report(y_test,y_pred)
print("Classification report:",result1)
```

```
Classification report:               precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

In [ ]: