Q1. Split the dataset into training dataset and testing dataset, with explanatory variables and target variables separated. Show the shapes of the 4 subsets.

A:

- **explanatory variables (=x)**
  - **training dataset shape: (149, 9)**
  - **testing dataset shape: (65, 9)**
- **target variables (=y)**
  - **training dataset shape: (149,) ≈ only 1 column(glass type) with 149 rows**
  - **testing dataset shape: (65,) ≈ only 1 column(glass type) with 65 rows**

```
(env) PS C:\Users\him91\OneDrive\桌面\5010推薦系統\Lab 1-2> python .\509557023.py
x_train shape: (149, 9)
y_train shape: (149,)
x_test shape: (65, 9)
y_test shape: (65,)
```

Q2. Try at least 3 different algorithms for classifying the glass types. Briefly describe what you did.

A:

- **Random Forest**
- **k-nearest neighbors**
- **DecisionTree**

Get the 30% of glass data to test, and the remained 70% data to training, then get the classification accuracy score as picture below.

```
RandomForestClassifier Accuracy: 81.54%
KNeighborsClassifier Accuracy: 67.69%
DecisionTreeClassifier Accuracy: 73.85%
```

Q3. Draw the confusion matrix and calculate the accuracy, precision, recall and F score on the testing data, for the method with the best performance you used in question 2 above.

A:

[confusion matrix]

1, 2, 3, 5, 6, 7 type numbers are mapped to 'building-float', 'building-non-float', 'vehicle-float', 'containers', 'tableware', 'headlamps'.

```
RandomForestClassifier Accuracy: 81.54%
                  building-float  building-non-float  vehicle-float  containers  tableware  headlamps
building-float                18                  3              0           0          0          0
building-non-float             5                 18              0           0          0          0
vehicle-float                  2                  0              3           0          0          0
containers                     0                  0              0           4          0          0
tableware                      0                  1              0           0          2          0
headlamps                      1                  0              0           0          0          8
```

[accuracy, precision, recall and F score]

	precision	recall	f1-score	support
1	0.70	0.90	0.79	21
2	0.82	0.78	0.80	23
3	1.00	0.60	0.75	5
5	1.00	0.75	0.86	4
6	1.00	0.67	0.80	3
7	1.00	0.89	0.94	9
accuracy			0.82	65
macro avg	0.92	0.77	0.82	65
weighted avg	0.84	0.82	0.82	65

Q4. (Bonus) What else did you do to try making the predictions better? (10%)

A:

- **Tried to change KNeighborsClassifier n_neighbors (default is 5.). After using GridSearchCV function to test each of numbers, I found that n_neighbors set to 4 can get the better accuracy than setting to 3 or 7.**
- **Tried to change RandomForestClassifier n_estimators(default is 100.). After using GridSearchCV function to test numbers range from 1 to 200, I found that n_estimators set to 25 can get the better accuracy(up to 86.15%) than setting to default.**
- **Tried to use GridSearchCV function mentioned before to find the best parameters of DecisionTreeClassifier. After testing criterion and splitter, the result shows the default setting is already the best setting.**