In [1]:

```
import numpy as np
```

In [2]:

```
#Q1 Create a NumPy ndarray named sales with the following values.
#It is assumed that the rows are for cities and the columns are for Quarters.
#Each value is the sales of vehicles for a specific city in a specific quarter.
sales=np.array([[1200, 1400, 800, 1100],
 [1300, 1500, 1600, 1000],
 [1100, 1200, 1000, 1050]])
```

In [3]:

```
sales
```

Out[3]:

```
array([[1200, 1400,  800, 1100],
       [1300, 1500, 1600, 1000],
       [1100, 1200, 1000, 1050]])
```

In [4]:

```
#Q2 . Create two NumPy array named cities and quarters with the following values.
#The order of the cities is used for the row order in Q2 and the order of Quarters is used f
or the column order in Q2.
cities=np.array(['Dallas', 'Houston', 'Austin'])
quarters=np.array(['Q1', 'Q2', 'Q3', 'Q4'])
```

In [5]:

```
cities
```

Out[5]:

```
array(['Dallas', 'Houston', 'Austin'], dtype='<U7')
```

In [6]:

```
quarters
```

Out[6]:

```
array(['Q1', 'Q2', 'Q3', 'Q4'], dtype='<U2')
```

In [32]:

```
#Q3
city=input("what is the city?")
row=np.where(cities==city)
array=[False]*3
index=row[0][0]
array[index]=True
print(array)
```

```
what is the city?Houston
[False, True, False]
```

In [33]:

```
#Q4 Write a statement to show the sales of all quarters of the input city.
#For example, if the input city is “Houston”, the output is array([[1300, 1500, 1600, 100
0]])
sales[city==cities]
```

Out[33]:

```
array([[1300, 1500, 1600, 1000]])
```

In [ ]:

```

```

In [13]:

```
#Q5 Write a statement to show the total sales of all quarters and also the average sales of
 each quarter of the input city.
#For example, if the input city is ‘Houston’, the output is 5400 and 1350.
tot_sale=sum(sales[row][0])
avg_sale=tot_sale/4
print(tot_sale)
print(avg_sale)
```

```
5400
1350.0
```

In [19]:

```
#Q6
Quarter = input("Which Quarter?")
print(Quarter)
```

```
Which Quarter?Q2
Q2
```

In [20]:

```
#Q7 Write a statement to display the sales of all cities in the input quarter. For example,
 if the input quarter is “Q2”
col=np.where(quarters==Quarter)
print(sales[:,col])
```

```
[[[1400]]

 [[1500]]

 [[1200]]]
```

In [22]:

```
#Q8 Find the largest sales in the input quarter. For example, if the input quarter is ‘Q2’
max_arr=np.amax(sales[:,col])
print(max_arr)
```

```
1500
```

In [25]:

```
#Q9 Find the largest sales in the input quarter. For example, if the input quarter is ‘Q2’
max_sales_cities=np.amax(sales, axis=1)
print(max_sales_cities)
```

```
[1400 1600 1200]
```

In [24]:

```
#Q10 Find the largest sales (in all cities) for each quarter.
max_sales_quarter=np.amax(sales, axis=0)
print(max_sales_quarter)
```

```
[1300 1500 1600 1100]
```

In [26]:

```
#Q11 Change the structure of sales as: use the rows for quarters and use the columns for cit
ies. Display sales.
sales_tr=sales.transpose()
print(sales_tr)
```

```
[[1200 1300 1100]
 [1400 1500 1200]
 [ 800 1600 1000]
 [1100 1000 1050]]
```