Given code:

```
import numpy as np
list_ = ['1', '2', '3', '4', '5']
array_list = np.array(object=list_)
```

Q1. Difference in data type of `list_` and `array_list`

- `list_` is a **Python list**, so its type is `list`.
- `array_list` is a **NumPy array**, so its type is `numpy.ndarray`.
- **Code to print data types:**

```
print("Data type of list_:", type(list_))
print("Data type of array_list:", type(array_list))
```

Expected output:

```
Data type of list_: <class 'list'>
Data type of array_list: <class 'numpy.ndarray'>
```

Q2. Data type of each element in `list_` and `array_list`

- In `list_`, each element is a **string**.
- In `array_list`, since we didn't specify `dtype`, NumPy infers `str` (string) as well.

Code:

```
# For list_
print("Data types of elements in list_:")
for elem in list_:
    print(type(elem))
```

```
# For array_list
print("\nData types of elements in array_list:")
for elem in array_list:
    print(type(elem))
```

Expected output:

```
Data types of elements in list_:
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>

Data types of elements in array_list:
<class 'numpy.str_'>
<class 'numpy.str_'>
<class 'numpy.str_'>
<class 'numpy.str_'>
<class 'numpy.str_'>
```

Note: NumPy uses `numpy.str_` for string elements.

Q3. Changing `array_list` dtype to int

```
array_list = np.array(object=list_, dtype=int)
```

- `list_` **still contains strings**.
- `array_list` **now contains integers**.

Code to print element types:

```
# For list_
print("Data types of elements in list_ after array_list change:")
for elem in list_:
    print(type(elem))

# For array_list
print("\nData types of elements in array_list after dtype=int:")
for elem in array_list:
    print(type(elem))
```

Expected output:

```
Data types of elements in list_ after array_list change:
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>

Data types of elements in array_list after dtype=int:
<class 'numpy.int64'>
<class 'numpy.int64'>
<class 'numpy.int64'>
<class 'numpy.int64'>
<class 'numpy.int64'>
```

✅ **Observation:** The Python list remains strings; NumPy array elements are now integers.

Next code:

```
num_list = [[1, 2, 3], [4, 5, 6]]
num_array = np.array(object=num_list)
```

Q4. Find `shape` and `size` of `num_array`

- `shape` gives the dimensions of the array.

- `size` gives the total number of elements.

Code:

```
print("Shape of num_array:", num_array.shape)
print("Size of num_array:", num_array.size)
```

Expected output:

```
Shape of num_array: (2, 3)
Size of num_array: 6
```

Q5. Create 3x3 array of zeros

Code:

```
zeros_array = np.zeros((3, 3))
print(zeros_array)
```

Expected output:

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

Size = 9, Shape = (3,3)

Q6. Create 5x5 identity matrix

- An identity matrix has `1`s on the diagonal, `0`s elsewhere.

Code:

```
identity_matrix = np.eye(5)
print(identity_matrix)
```

Expected output:

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```