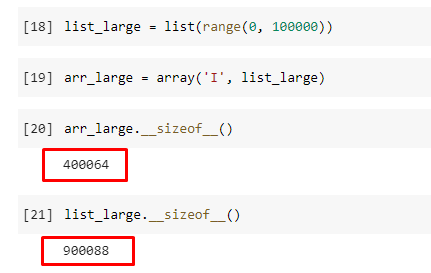
Q1. What are the benefits of the built-in array package, if any?

1.When dealing with a huge number of elements, arrays use less memory than lists and they offer faster execution than lists.



2.The size of the array is not fixed in Python. Hence, we need not specify how many elements we are going to store into an array in the beginning.

3.Arrays can grow or shrink in memory dynamically (during runtime).

4.Arrays are useful to handle a collection of elements like a group of numbers or characters.

5.You do not need to worry about the allocation of memory when creating an array, as all elements are allocated memory in contiguous memory locations of the array. There isn't any chance of extra memory being allocated if there is an array. This avoids the issue of overflow or shortage of memory.

Q2. What are some of the array package's limitations?

An array has a fixed size which means you cannot add/delete elements after creation. You also cannot resize them dynamically.

* Unlike lists in Python, cannot store values of different data types in a single array.
* When inserting into an array, the insertion process requires moving each element from its original location to the next available slot. The shift cost increases linearly with the length of the array.
* Deleting an item from an array involves copying every preceding element to fill up the gap left behind by the deleted element. Deleting items from an array is very expensive because of this reason.

Q3. Describe the main differences between the array and numpy packages.

The array package doesn't provide any help with numerical calculation with the items insdie it in number form while NumPy give you a wide variety of numerical operations.

An array is a single dimensional entity which hold the numerical data, while numpy can have more than 1 dimension.

In case of array, item can be accessed by its index position and it is easy task while in numpy item is accessed by its column and row index, which makes it slightly time taking. Same goes with appending operation.

Q4. Explain the distinctions between the empty, ones, and zeros functions.

The distinctions between the empty, ones, and zero functions are as follows :

* Empty function: An empty function is a function that does not contain any statement within its body. If you try to write a function definition without any statement in python ,it will return an error. To avoid this, we use pass statement. pass is a special statement in Python that does nothing. It only works as a dummy statement.
* Ones: This function returns a new array of given shape and data type, where the element’s value is 1.
* Zeros: This function returns a new array of given shape and data type, where the element’s value is 0.

Q5. In the fromfunction function, which is used to construct new arrays, what is the role of the callable argument?

**numpy.fromfunction(*function*, *shape*, *\**, *dtype=<class 'float'>*, *like=None*, *\*\*kwargs*)**[**[source]**](https://github.com/numpy/numpy/blob/v1.24.0/numpy/core/numeric.py#L1798-L1866)

**function*callable***

The function is called with N parameters, where N is the rank of [**shape**](https://numpy.org/doc/stable/reference/generated/numpy.shape.html#numpy.shape). Each parameter represents the coordinates of the array varying along a specific axis. For example, if [**shape**](https://numpy.org/doc/stable/reference/generated/numpy.shape.html#numpy.shape) were (2, 2), then the parameters would be array([[0, 0], [1, 1]]) and array([[0, 1], [0, 1]])

**np.fromfunction(lambda i, j: j, (2, 2), dtype=float)**

***array([[0., 1.],***

***[0., 1.]])***

Q6. What happens when a numpy array is combined with a single-value operand (a scalar, such as an int or a floating-point value) through addition, as in the expression A + n?

 If any scaler value such as integer is added to the numpy array then all the elements inside the array will add that value in it.

Q7. Can array-to-scalar operations use combined operation-assign operators (such as += or \*=)? What is the outcome?

It will carry out provided operation on all elements of array.

Q8. Does a numpy array contain fixed-length strings? What happens if you allocate a longer string to one of these arrays?

 Yes, it is possible that we can include a string of fixed length in numpy array. The dtype of any numpy array containing string values is the maximum length of any string present in the array.Once set, it will only be able to store new string having length not more than the maximum length at the time of the creation. If we try to reassign some another string value having length greater than the maximum length of the existing elements, it simply discards all the values beyond the maximum length accept upto those values which are under the limit.

Q9. What happens when you combine two numpy arrays using an operation like addition (+) or multiplication (\*)? What are the conditions for combining two numpy arrays?

It will simply add or multiply element to element at same position.

**conditions for combining two numpy arrays :**

When operating on two arrays, NumPy compares their shapes element-wise. It starts with the trailing (i.e. rightmost) dimension and works its way left. Two dimensions are compatible when

1. they are equal, or
2. one of them is 1.

Q10. What is the best way to use a Boolean array to mask another array?

The best way to use a Boolean array to mask another array is by Using masked\_where of numpy package.

Q11. What are three different ways to get the standard deviation of a wide collection of data using both standard Python and its packages? Sort the three of them by how quickly they execute.

The statistics.stdev() method calculates the standard deviation from a sample of data.this methods present in statistics module of python.

The statistics.pstdev() method calculates the standard deviation from an entire population.

**np.std** and math package can be used individually.

Q12. What is the dimensionality of a Boolean mask-generated array?

 It will have same dimensionality as input array.