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

The Array is built-in Python, which only means that you don't need to download and install it because it comes with native Python for sure.

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

**An array which is formed will be homogeneous**. That is, in an integer array only integer values can be stored, while in a float array only floating value and character array can have only characters. Thus, no array can have values of two data types

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

Built-in array module defines an object type which can efficiently represent an array of basic values: characters, integers, floating point numbers. Arrays are sequence types and behave very much like lists, except that the type of objects stored in them is constrained. This means objects stored in the array are of a homogeneous(same) type.

Numpy module in python is generally used for matrix and array computations. While using the numpy module, built-in function ‘array’ is used to create an array. A prototype of array function is where everything is optional except object. And the object is an array, any object exposing the array interface, an object whose *\_\_array\_\_* method returns an array or any (nested) sequence. And dtype is desired data type for array.

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

zeros() function is used to create an array based on the particular shape and type. All array elements are initialized to 0, which is created by the zeros() function. ones() function works like the zeros() function. But the elements of the array created by the ones() function are initialized to  1.

Just like numpy.zeros(), the numpy.empty() function doesn't set the array values to zero, and it is quite faster than the numpy.zeros().

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

Callable function is called with N parameters, where N is the rank of shape. **Each parameter represents the coordinates of the array varying along a specific axis**. shape : [(N, ) tuple of ints] Shape of the output array, which also determines the shape of the coordinate arrays passed to function.

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?

It will add that single-value operand to each array element.

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

Yes.

These operators are used in place of the usual assignment operator, and first perform the selected binary operation, and then re-assign the resulting value to the variable on the left-hand side of the assignment operator. The following examples show the equivalence of assignment operators with the usual assignment and binary operators.

$x += 5; # equivalent to $x = $x + 5

$value \*= 2; # equivalent to $value = $value \* 2

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

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.**

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?

Scalars can be added and subtracted from arrays and arrays can be added and subtracted from each other:

NumPy array can be multiplied by each other using matrix multiplication. These matrix multiplication methods include element-wise multiplication, the dot product, and the cross product.

Join a sequence of arrays along an existing axis. **The arrays must have the same shape, except in the dimension corresponding to axis** (the first, by default). The axis along which the arrays will be joined. If axis is None, arrays are flattened before use.

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

* import numpy as np import numpy.ma as ma
* arr = np.zeros(7) print("Array...", arr) ...
* print("Masked Array...", ma.make\_mask(arr)) ...
* print("Array type...", arr.dtype) ...
* print("Array Dimensions...",arr.ndim) ...
* print("Our Array Shape...",arr.shape) ...
* print("Elements in the Array...",arr.size)

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.

**Mean, Median, Mode**

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

**One-dimensional**