Q1. Describe the differences between text and binary files in a single paragraph.

In text file, text, character, numbers are stored one character per byte i.e. 32667 occupies 5 bytes even though it occupies 2 bytes in memory. In binary file data is stored in binary format and each data would occupy the same number of bytes on disks as it occupies in memory.

Q2. What are some scenarios where using text files will be the better option? When would you like to use binary files instead of text files?

While a small error in a binary file may make it unreadable, a small error in a text file may simply show up once the file has been opened.

Text files are more restrictive than binary files since they can only contain textual data. However, unlike binary files, they are less likely to become corrupted

Q3. What are some of the issues with using binary operations to read and write a Python integer directly to disc?

While writing to the file we need to use the encode method for wirting the data into the bytes format.

Also while retrieving the data we need to specify or convert the data to integer.

Q4. Describe a benefit of using the with keyword instead of explicitly opening a file.

The withstatement in Python is used for resource management and exception handling. You’d most likely find it when working with file streams. For example, the statement ensures that the file stream process doesn’t block other processes if an exception is raised, but terminates properly.

Q5. Does Python have the trailing newline while reading a line of text? Does Python append a newline when you write a line of text?

Python readline() is a file method that helps to read one complete line from the given file. It has a trailing newline (“\n”) at the end of the string returned.

Yes python appends the newline while writing.

Q6. What file operations enable for random-access operation?

With the seek function we can reach out to the particular file offset or index and from that we can read the file randomly.

Q7. When do you think you'll use the struct package the most?

The struct module in Python is used to convert native Python data types such as strings and numbers into a string of bytes and vice versa. What this means is that users can parse binary files of data stored in C structs in Python.

It is used mostly for handling binary data stored in files or from network connections, among other sources.

Q8. When is pickling the best option?

If we don't need a human-readable format or a standard interoperable format, or if you need to serialize custom objects, then can go with pickle

Q9. When will it be best to use the shelve package?

The shelve module in Python's standard library is a simple yet effective tool for persistent data storage when using a relational database solution is not required. The shelf object defined in this module is dictionary-like object which is persistently stored in a disk file

Q10. What is a special restriction when using the shelve package, as opposed to using other data dictionaries?

The shelf dictionary has certain restrictions. Only string data type can be used as key in this special dictionary object,