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

Ans: Text files are organized around lines, each of which ends with a newline character ('\n'). The source code files are themselves text files. A binary file is the one in which data is stored in the file in the same way as it is stored in the main memory for processing.

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?

Ans: A binary file is usually very much smaller than a text file that contains an equivalent amount of data. For image, video, and audio data this is important. Small files save storage space, can be transmitted faster, and are processed faster. I/O with smaller files is faster, too, since there are fewer bytes to move.

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

Ans: To open a file in binary format, add 'b' to the mode parameter. Hence the "rb" mode opens the file in binary format for reading, while the "wb" mode opens the file in binary format for writing.

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

Ans: It makes resource management safer.

Cleans up the code compared to try… finally.

Helps in avoiding resource leaks.

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?

Ans: The text file example. txt has two lines inside it, but the readline() method only reads one line from the file and returns it. The readline() method also adds a trailing newline character at the end of the string.

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

Ans: e learned about the functions that help us achieve random access in files, including the fseek() function, which helps us send a file pointer to a specified location; the ftell() function, which is used to find out the exact position of the file pointer with respect to the beginning; and the rewind() function

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

Ans: It is used mostly for handling binary data stored in files or from network connections, among other sources. This process needs to be done at the start of the program.

Q8. When is pickling the best option?

Ans: An untrusted client or an untrusted server can cause remote code execution. Thus pickle should never be used between unknown parties. Ensure the parties exchanging pickle have an encrypted network connection. This prevents alteration or replay of data on the wire.

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

Ans: a simple yet effective tool for persistent data storage when using a relational database solution is not required.

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