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

Ans

* Binary files typically contain a sequence of bytes, or ordered groupings of eight bits.
* When creating a custom file format for a program, these bytes are arranged into a format that

stores the necessary information for the application.

* Binary file formats may include multiple types of data in the same file, such as image, video, and audio data.
* This data can be interpreted by supporting programs, but will show up as garbled text in a text editor.
* 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.

* 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 may be saved in either a plain text (.TXT) format and rich text (.RTF) format. A typical plain text file contains several lines of text that are each followed by an End-of-Line (EOL) character.
* An End-of-File (EOF) marker is placed after the final character, which signals the end of the file. Rich text files use a similar file structure, but may also include text styles, such as bold and italics, as well as page formatting information

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 Text files include small size and versatility. Kilobytes or megabytes smaller than the same data

stored in other formats, they can be rapidly and massively exchanged via email or disk. Most can be opened on computers running diverse operating systems, using very basic software.

Binary files is that they are more efficient. In terms of memory, storing values using numeric formats, rather than as text characters, tends to use less memory. In addition, binary formats also offer advantages in terms of speed of access.

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

Ans While reading, the read method returns a sequence of bytes as a string. With using binary operations i.e. using bitwise operator the integers are first converted into binary and then operations are performed on bit by bit. This is a very lengthy task from the usual steps. More over it is not possible for a reader to read the integers in binary format.

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

Ans When a file is opened using the 'with' keyword, if some exceptions occur after opening a file, or at the end of the file, it automatically does the closing of the file. Thereby not leaving a file in open mode and there would no need to explicitly close a file.

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 While reading a newline of text from a text file, python reads the newline also.While writing the python doesnt append a new line at end of line. It has to be handled explicitly.

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

Ans The file operations like seek(pos,orig), tell() enable random access operations.

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

Ans Whenever suppose we need to convert the data types (like string and number) into string of bytes we use **struct** module. When we need to parse the binary files of data stored in C into python we generally used this module

Q8. When is pickling the best option?

Ans Pickling is best option when creating a new binary file using python.

#Once a file is created it can be read by other python programs. The functions available in the pickle package

# take care how to represent the data is written in the file.

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

Ans When we need to store the data in a constant/continuous manner such that we do not require the relational database, then we use shelve package. The shelf object defined in this module is a dictionary-like object which is persistently stored in a disk file. This creates a file similar to dbm database. Only string data type can be used as key in this special dictionary object.

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

Ans.One Restriction of shelve objects is that the keys to the objects must be strings, but the values stored in a shelve object can be any Python object, as long as it can be written with the pickle module