1. What advantages do Excel spreadsheets have over CSV spreadsheets?

**Answer**: Excel is superior to CSV spreadsheets as follows :-

1) Excel can perform VBA operations, functions, formulas which CSV doesnt natively support
2) Reading and understanding large data set is easier on Excel then on CSV
3) Excel provides linking external data sets and sources, which CSV does not allow.

2.What do you pass to csv.reader() and csv.writer() to create reader and writer objects?

**Answer**: csv.reader(csvfile, delimiter='', **fmtparams)

Where csvfile is the full path of the csv file, fmtparams is the formatting parameters, and delimiter specifies the boundary between words, typical a comma for csv

csv.writer(csvfile, delimiter='', **fmtparams)

The same can be passed to csv.writer() which creates a writer object which converts the users data into delimited strings.

3. What modes do File objects for reader and writer objects need to be opened in?

**Answer**: To open a file for reading, we have to use the 'r' mode, which need not be specified as 'r' mode is default for csv.reader()

To create a writer object, we have to use the 'w' mode, This mode opens a new file, or clears the existing file if mentioned and truncates it from the beginning.

4. What method takes a list argument and writes it to a CSV file?

**Answer**: to_csv() function takes a list and converts it to a CSV file. to_csv() is part of the pandas library.

Or we can also use the csv.writer() method of the csv module.

```
import csv RESULT = ['apple','cherry','orange','pineapple','strawberry']
with open('output.csv','w') as result_file:
    wr = csv.writer(result_file, dialect='excel')
    wr.writerow(RESULT)

#Here we don't have to pass ‘w’ mode to writer() function as the file object is already opened with ‘w’ mode
```

5. What do the keyword arguments delimiter and line terminator do?

**Answer**: The line terminator is the character that comes at the end of the row, by default line terminator is a new line.

Delimiter argument separates each cell or word set in the file, for instance **delimiter=’\t’** will separate each cell by a tab.

**lineterminator='\n\n'** will separate each new line by two vertical spaces

6. What function takes a string of JSON data and returns a Python data structure?

**Answer**: **decode(o)  or  json.loads()** method return Python data structure of JSON string or data.

```
import json
from json.decoder import JSONDecoder
new_string='{"color": ["red","yellow"]}'

JSONDecoder().decode(new_string)

OUTPUT >>> {'color': ['red', 'yellow']}
```

7. What function takes a Python data structure and returns a string of JSON data?

**Answer**: **json.dumps()** or **encode(o),**

both takes python data structure and returns JSON string of the python data