1. Set the variable test1 to the string 'This is a test of the emergency text system,' and save test1 to a file named test.txt.

Ans: test1 **=** 'This is a test of the emergency text system'

len(test1)

Out[1]:

43

In [5]:

**with** open('test.txt', 'wt') **as** outfile:

outfile**.**write(test1)

In [6]:

outfile**.**close()

2. Read the contents of the file test.txt into the variable test2. Is there a difference between test 1 and test 2?

Ans:

**with** open('test.txt', 'rt') **as** infile:

test2 **=** infile**.**read()

len(test2)

Out[8]:

43

In [9]:

test1 **==** test2

Out[9]:

True

3. Create a CSV file called books.csv by using these lines:

title,author,year

The Weirdstone of Brisingamen,Alan Garner,1960

Perdido Street Station,China Miéville,2000

Thud!,Terry Pratchett,2005

The Spellman Files,Lisa Lutz,2007

Small Gods,Terry Pratchett,1992

Ans:

text **=** '''title,author,year

The Weirdstone of Brisingamen,Alan Garner,1960

Perdido Street Station,China Miéville,2000

Thud!,Terry Pratchett,2005

The Spellman Files,Lisa Lutz,2007

Small Gods,Terry Pratchett,1992

'''

**with** open('books.csv', 'wt') **as** outfile:

outfile**.**write(text)

4. Use the sqlite3 module to create a SQLite database called books.db, and a table called books with these fields: title (text), author (text), and year (integer).

Ans:

**import** sqlite3

db **=** sqlite3**.**connect('books.db')

curs **=** db**.**cursor()

curs**.**execute('''create table book (title text, author text, year int)''')

db**.**commit()

5. Read books.csv and insert its data into the book table.

**Ans:**

**import** csv

**import** sqlite3

ins\_str **=** 'insert into book values(?, ?, ?)'

**with** open('books.csv', 'rt') **as** infile:

books **=** csv**.**DictReader(infile)

**for** book **in** books:

curs**.**execute(ins\_str, (book['title'], book['author'], book['year']))

db**.**commit()

6. Select and print the title column from the book table in alphabetical order.

Ans:

sql **=** 'select title from book order by title asc'

**for** row **in** db**.**execute(sql):

print(row)

('Perdido Street Station',)

('Small Gods',)

('The Spellman Files',)

('The Weirdstone of Brisingamen',)

('Thud!',)

In [16]:

*#to print the title value without that tuple stuff (parentheses and comma):*

**for** row **in** db**.**execute(sql):

print(row[0])

Perdido Street Station

Small Gods

The Spellman Files

The Weirdstone of Brisingamen

Thud!

7. From the book table, select and print all columns in the order of publication.

Ans:

**for** row **in** db**.**execute('select \* from book order by year'):

print(row)

('The Weirdstone of Brisingamen', 'Alan Garner', 1960)

('Small Gods', 'Terry Pratchett', 1992)

('Perdido Street Station', 'China Miéville', 2000)

('Thud!', 'Terry Pratchett', 2005)

('The Spellman Files', 'Lisa Lutz', 2007)

In [18]:

*#To print all the fields in each row, just separate with a comma and space:*

**for** row **in** db**.**execute('select \* from book order by year'):

print(**\***row, sep**=**', ')

o/p

The Weirdstone of Brisingamen, Alan Garner, 1960

Small Gods, Terry Pratchett, 1992

Perdido Street Station, China Miéville, 2000

Thud!, Terry Pratchett, 2005

The Spellman Files, Lisa Lutz, 2007

8. Use the sqlalchemy module to connect to the sqlite3 database books.db that you just made in exercise 6.

Ans: **import** sqlalchemy

conn **=** sqlalchemy**.**create\_engine('sqlite:///books.db')

sql **=** 'select title from book order by title asc'

rows **=** conn**.**execute(sql)

**for** row **in** rows:

print(row)

('Perdido Street Station',)

('Small Gods',)

('The Spellman Files',)

('The Weirdstone of Brisingamen',)

('Thud!',)

9. Install the Redis server and the Python redis library (pip install redis) on your computer. Create a Redis hash called test with the fields count (1) and name ('Fester Bestertester'). Print all the fields for test.

Ans;

**import** redis

conn **=** redis**.**Redis()

conn**.**delete('test')

Out[3]:

0

In [11]:

conn**.**hmset('test', {'count': 1, 'name': 'Fester Bestertester'})

:1: DeprecationWarning: Redis.hmset() is deprecated. Use Redis.hset() instead.

conn.hmset('test', {'count': 1, 'name': 'Fester Bestertester'})

Out[11]:

True

In [12]:

conn**.**hgetall('test')

Out[12]:

{b'count': b'1', b'name': b'Fester Bestertester'}

10. Increment the count field of test and print it.

Ans:

conn**.**hincrby('test', 'count', 3)

Out[13]:

4

In [14]:

conn**.**hget('test', 'count')

Out[14]:

b'4'