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:

```
In [4]:
test1='This is a test of the emergency text system'
print(test1)
with open('test.txt','w') as file:
    file.write(test1)
    file.close()
```

```
This is a test of the emergency text system
```

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

Ans:

```
In [5]:
with open('test.txt','r') as file:
    test2 = file.read()
print(test2)
print(test1 == test2)
```

```
This is a test of the emergency text system
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:

```
In [9]:
book = '''title,author,year \n
The Weirdstone of Brisingamen,Alan Garner,1960 \n
Perdido Street Station,China Miéville,2000 \n
Thud!,Terry Pratchett,2005 \n
The Spellman Files,Lisa Lutz,2007 \n
Small Gods,Terry Pratchett,1992'''
with open('books.csv','w') as file:
    file.write(book)
    file.close()
! type books.csv
```

```
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
```

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:

```
In [10]:
import sqlite3
db = sqlite3.connect('books.db')
cursor = db.cursor()
cursor.execute('CREATE TABLE books (title text, author text, year int)')
db.commit()
db.close()
```

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

Ans:

In [16]:
```
import csv
conn = sqlite3.connect("books.db")
cursor = conn.cursor()
with open("books.csv","r") as file:
    books = csv.DictReader(file)
    for ele in books:
        cursor.execute("INSERT INTO books VALUES (?,?,?)",
                       (ele['title'],ele['author'],ele['year']))
conn.commit()
conn.close()
```

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

Ans:

In [6]:
```
import sqlite3
conn = sqlite3.connect('books.db')
cursor = conn.cursor()
output = cursor.execute("SELECT title FROM books ORDER BY title ASC")
for ele in output:
    print(ele[0])
conn.commit()
conn.close()
```

```
Perdido Street Station
Small Gods
The Spellman Files
The Weirdstone of Brisingamen
Thud!
```

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

Ans:

In [10]:
```
import sqlite3
conn = sqlite3.connect('books.db')
cursor = conn.cursor()
ouput = cursor.execute("SELECT * FROM books ORDER BY year")
for record in ouput:
    print(record)
```

```
('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:

In [11]:
```
import sqlalchemy
conn = sqlalchemy.create_engine('sqlite:///books.db')
conn
```

Out[11]:
```
Engine(sqlite:///books.db)
```

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:

In [12]:
```
! python -m pip install redis
```

```
Collecting redis
  Downloading redis-4.2.2-py3-none-any.whl (226 kB)
Collecting deprecated>=1.2.3
  Downloading Deprecated-1.2.13-py2.py3-none-any.whl (9.6 kB)
Collecting async-timeout>=4.0.2
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Requirement already satisfied: packaging>=20.4 in c:\users\thejaswini\anaconda3\lib\site-packages (from redis) (21.0)
Requirement already satisfied: wrapt<2,>=1.10 in c:\users\thejaswini\anaconda3\lib\site-packages (from deprecated>=1.2.3->redi
s) (1.12.1)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\thejaswini\anaconda3\lib\site-packages (from packaging>=20.4->redi
s) (3.0.4)
Installing collected packages: deprecated, async-timeout, redis
Successfully installed async-timeout-4.0.2 deprecated-1.2.13 redis-4.2.2
```

In [8]:

```
conn = redis.StrictRedis(host='localhost',port=6379,db=0)
conn.hset("Test", "Count", "1")
conn.hset("Test", "Name", "Fester Bestertester")
print("The keys present in the Redis hash:");
print(conn.hkeys("Test"))
print("The values present in the Redis hash:");
print(conn.hvals("Test"))
print("The keys and values present in the Redis hash are:")
print(conn.hgetall("Test"))
```

```
The keys present in the Redis hash:
[b'Count', b'Name']
The values present in the Redis hash:
[b'1', b'Fester Bestertester']
The keys and values present in the Redis hash are:
{b'Count': b'1', b'Name': b'Fester Bestertester'}
```

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

Ans:

In [10]:

```
conn.hincrby("test", "count", 1)
conn.hget("test", "count")
```

Out[10]:

```
b'2'
```