“Optional” Lab 6: Putting All Together

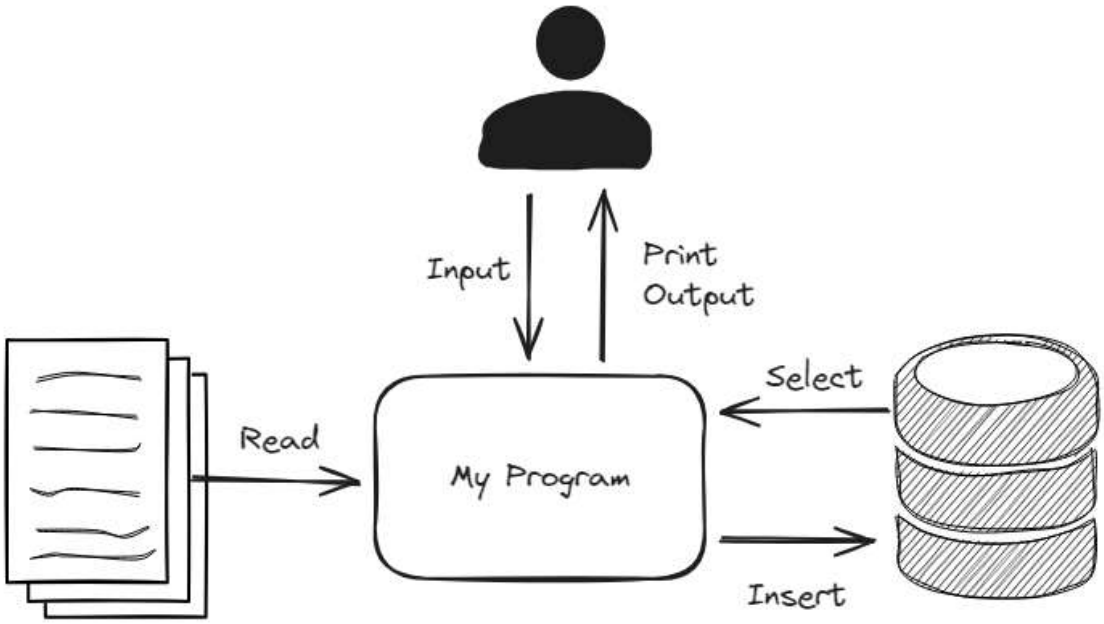
Overview

In this lab we will put everything we learnt together in one program.

- In Lab 1, we learned how to get **input** and print **output**
- In Lab 2, we knew how to use **if** to check some conditions
- In Lab 3, we wrote repeated logic using **loops**
- In Lab 4, we **read data** from text files
- In Lab 5, we worked with **database**

Requirement

You are required to read data from a text file, and insert it into a database, then based on user input you will execute a database query to retrieve information about the data that you had inserted.



Steps

1. Copy the following December weather data into a file and save it to your Desktop with name **weather.txt**. The file has four columns: date, day, temperature and weather status.

```
01-Dec   Fri   20   Cloudy
02-Dec   Sat   25   Cloudy
03-Dec   Sun   20   Cloudy
04-Dec   Mon   19   Sunny
05-Dec   Tue   25   Cloudy
06-Dec   Wed   25   Rains
07-Dec   Thu   20   Clear
08-Dec   Fri   17   Sunny
09-Dec   Sat   19   Sunny
10-Dec   Sun   22   Sunny
11-Dec   Mon   17   Sunny
12-Dec   Tue   24   Cloudy
13-Dec   Wed   24   Rains
14-Dec   Thu   21   Rains
15-Dec   Fri   23   Rains
16-Dec   Sat   23   Cloudy
17-Dec   Sun   23   Rains
18-Dec   Mon   23   Cloudy
19-Dec   Tue   17   Sunny
20-Dec   Wed   19   Sunny
21-Dec   Thu   21   Sunny
22-Dec   Fri   17   Sunny
23-Dec   Sat   25   Cloudy
24-Dec   Sun   20   Cloudy
25-Dec   Mon   18   Rains
26-Dec   Tue   22   Rains
27-Dec   Wed   23   Clear
28-Dec   Thu   21   Clear
29-Dec   Fri   20   Sunny
30-Dec   Sat   24   Clear
31-Dec   Sun   18   Clear
```

2. Open SQLite Browser, and create a database using the **“New Database”** button and use the name **lab6**. Then create a table named **“weather”** with four columns:
   - **DATE** with type TEXT
   - **DAY** with type TEXT
   - **TEMPERATURE** with type INTEGER
   - **STATUS** with type TEXT

   Warning: Don’t forget to press the **“Write Changes”** button to save your database

3. Copy the following code that reads the data from the file, and insert it into the database. **Warning**: Change the two paths to match the location of the file and the database at your machine

```
import sqlite3

name = “C:/Users/moham/Desktop/weather.txt”
handle = open(name, 'r')

try:
  database = “C:/Users/moham/Desktop/lab6.db”
  connection = sqlite3.connect(database)
  cursor = connection.cursor()
  print("Successfully Connected to SQLite")

  delete_query = "DELETE FROM weather"
  cursor.execute(delete_query)

  for line in handle:
     columns = line.split()

     date = columns[0]
     day = columns[1]
     temp = columns[2]
     status = columns[3]

     query="INSERT INTO weather VALUES('"+date+"','"+day+"',"+temp+",'"+status+"')"
     cursor.execute(query)

     print("Successfully inserted weather data of: ", date)

  cursor.close()
  connection.close()

except sqlite3.Error as error:
  print("Error while connecting to SQLite", error)
  connection.close()
```

Hint: Don’t waste your time rewriting the program, you can copy/paste the program from the part above it is not an image 🙂

Note: the program above inserts the data to the database every time you run the program. That is not what happens in real life where data is inserted only once. That is why in this program we delete the data before we try to insert it.

Exercise:

Modify your program to read the DATE as an input from the user, and search in the database for the weather information of that DATE. If the data was found then print it to the user, otherwise you should print that there is no data available.

The program should act as the following when there is data:

```
Please enter the date:23-Dec
The weather at that date is:
 [('23-Dec', 'Sat', 25, 'Cloudy')]
```

And as the following when there is no data

```
Please enter the date:15-Nov
There is no data available
```

Hint: the query that search in the database should be something like the following
`SELECT * FROM weather WHERE DATE = '23-Dec'`

Hint: you can use the `len()` function with the variable named "records" to know how many records are returned from the database.