**Ex.11 Quiz management system**

**Question:**

1. Develop a quiz application with two categories: Quiz Master and Student. The quiz master should develop the quiz by storing “N” number of questions with “3” options for each question and the correct answer for the question should also be stored. The questions, options and the correct answer should be stored using appropriate data structure. Display the generated questions and their options. Maintain the students list using appropriate data structure. The questions should be displayed randomly from the master’s question set along with their options to the students. The students will be selecting their options. The answers should be compared with correct answers and for each correct answer, 5 marks should be added and for each incorrect answer, 1 mark will be deducted. Display the student’s score at the end of the quiz. A score board should be maintained which stores the marks of each student. Order the students based on the marks and display it.

Assignment 1

1. For the above problem, identify the modules required for building the entire system by making use of decomposition techniques.
2. Develop the algorithms for the identified modules with the required inputs and outputs.
3. Draw a flowchart depicting the working of the student module by)

Assignment 2

1. Develop the user-defined incorporating the required modules. functions for the modules designed in Assignment 1.)

(ii) Develop the quiz master module, student module as a working model by invoking the functions.

1. Make use of conditional statements and looping constructs wherever possible
2. Make use of suitable data structures to store and process the application.

**Aim**

The aim of this code is to create a python quiz management system using files which has a different menu for the student and the quiz master

**Code:**

**def main():**

**global marks**

**global ques**

**global name**

**global value**

**name=input("Enter your name")**

**while len(value)>0:**

**value=list(value)**

**i=random.choice(range(0,len(value)))**

**print(ques,value[i][0],value[i][1])**

**ans=input("Enter the answer")**

**if ans==value[i][2][0]:**

**marks+=4**

**ques+=1**

**else:**

**marks-=1**

**ques+=1**

**value.pop(i)**

**print("You have completed the quiz")**

**print(name,marks, ques,"Questions attempted")**

**F=open("high\_scores.dat","rb")**

**record=pickle.load(F)**

**record.append([name,marks])**

**record.sort(key=lambda record : record [1],reverse=True)**

**F.close()**

**F=open("high\_scores.dat","wb")**

**pickle.dump(record,F)**

**F.close()**

**def add\_ques():**

**F=open("quiz.dat","rb")**

**value=pickle.load(F)**

**n=int(input("How many questions do you want to enter?"))**

**for i in range(n):**

**que=input("Enter the question")**

**opt=input("enter the options")**

**ans=input("enter the answer")**

**value.append([[que],[opt],[ans]])**

**F.close()**

**F=open("quiz.dat","wb")**

**pickle.dump(value,F)**

**print("Questions added successfully")**

**F.close()**

**import random**

**import pickle**

**F=open("quiz.dat","rb")**

**value=pickle.load(F)**

**#value=value.split("-")**

**n=len(value)**

**print("there are ",n ,"questions")**

**F.close()**

**marks=0**

**ques=0**

**ch=input("Are you the quiz master or the student q/s")**

**while True and ch in "sS":**

**choice=input("do you want to Start quiz? yes/no \n view high scores? h")**

**if choice in ["yes","Yes"]:**

**main()**

**elif choice in ["h","H"]:**

**F=open("high\_scores.dat","rb")**

**value=pickle.load(F)**

**print(value)**

**F.close()**

**else:**

**break**

**while True and ch in "qQ":**

**choice=input("Do you want to add more questions? add /view high scores? h or view questions? v")**

**if choice in ["h","H"]:**

**F=open("high\_scores.dat","rb")**

**value=pickle.load(F)**

**print(value)**

**F.close()**

**elif choice in ["add","Add"]:**

**add\_ques()**

**elif choice=="v":**

**F=open("quiz.dat","rb")**

**value=pickle.load(F)**

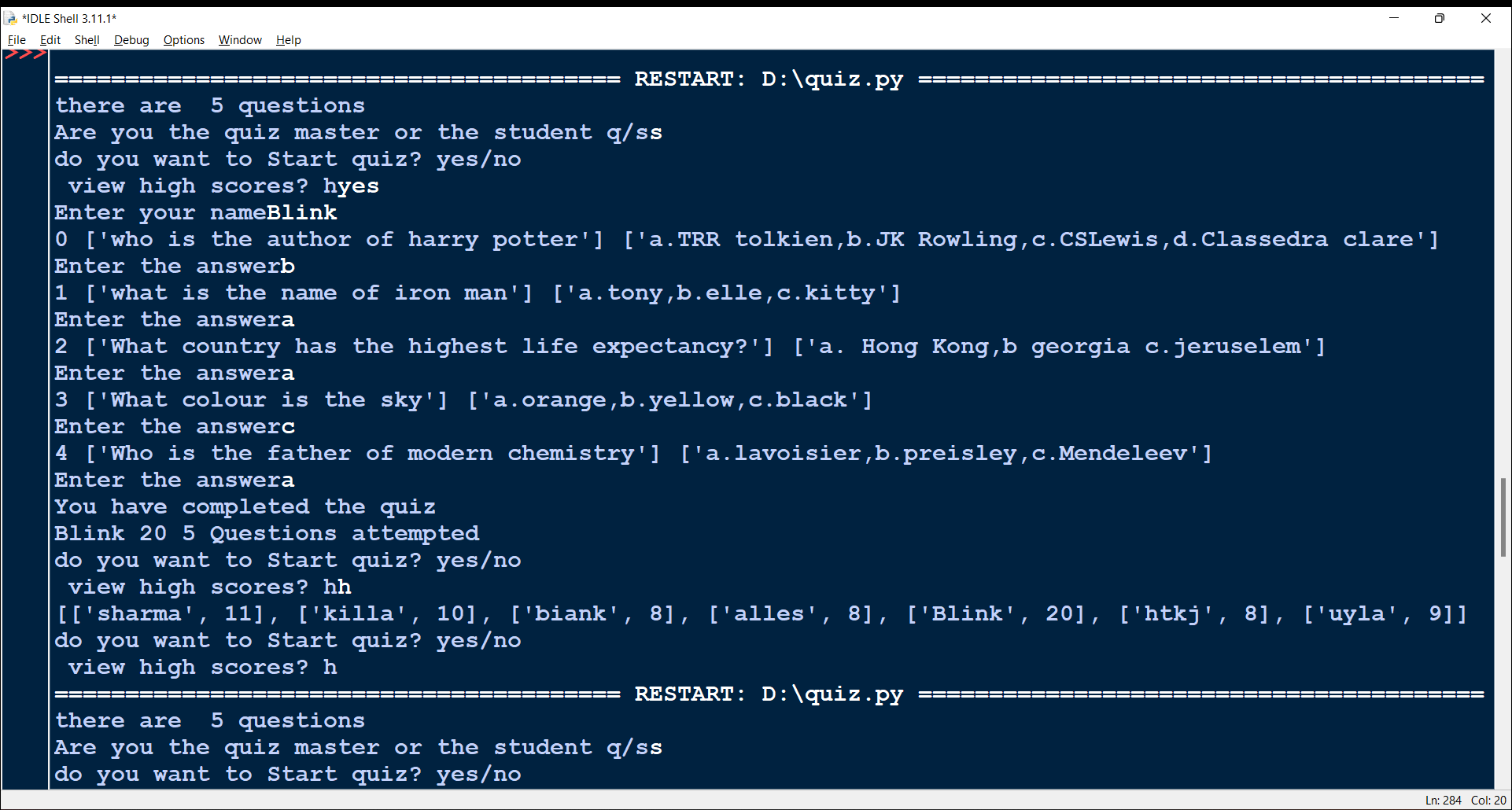
**print(value)**

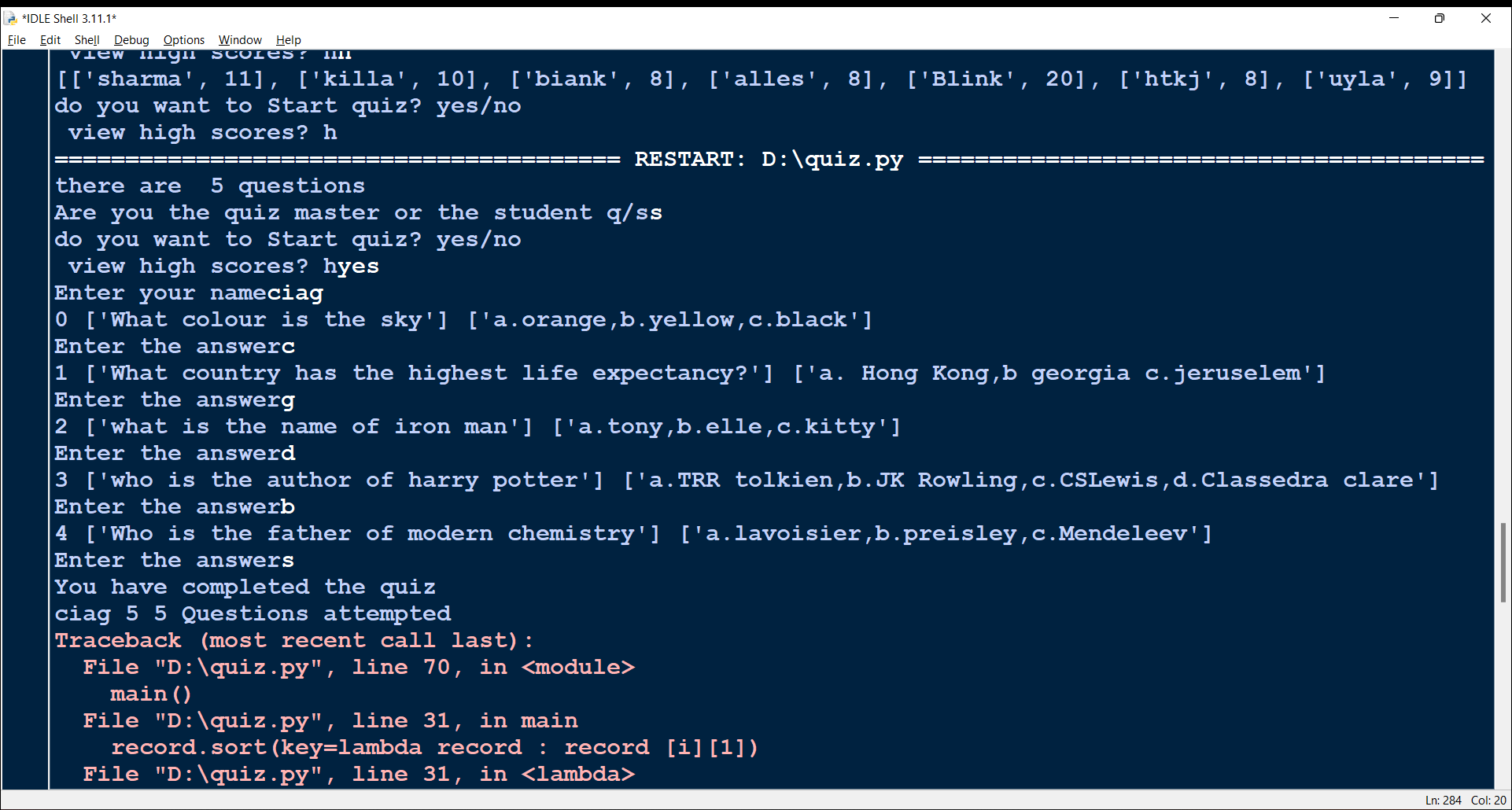
**else:**

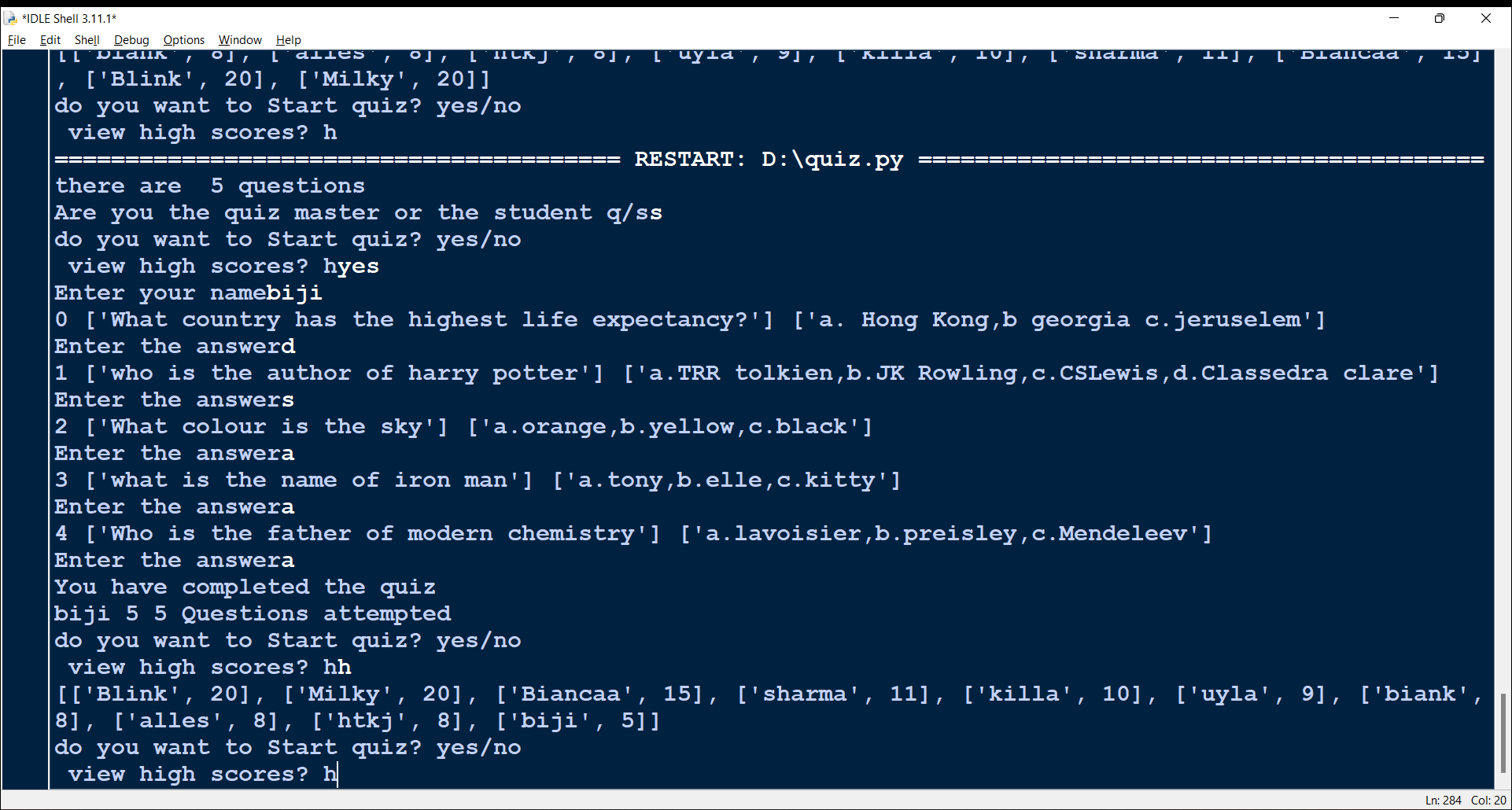
**break**

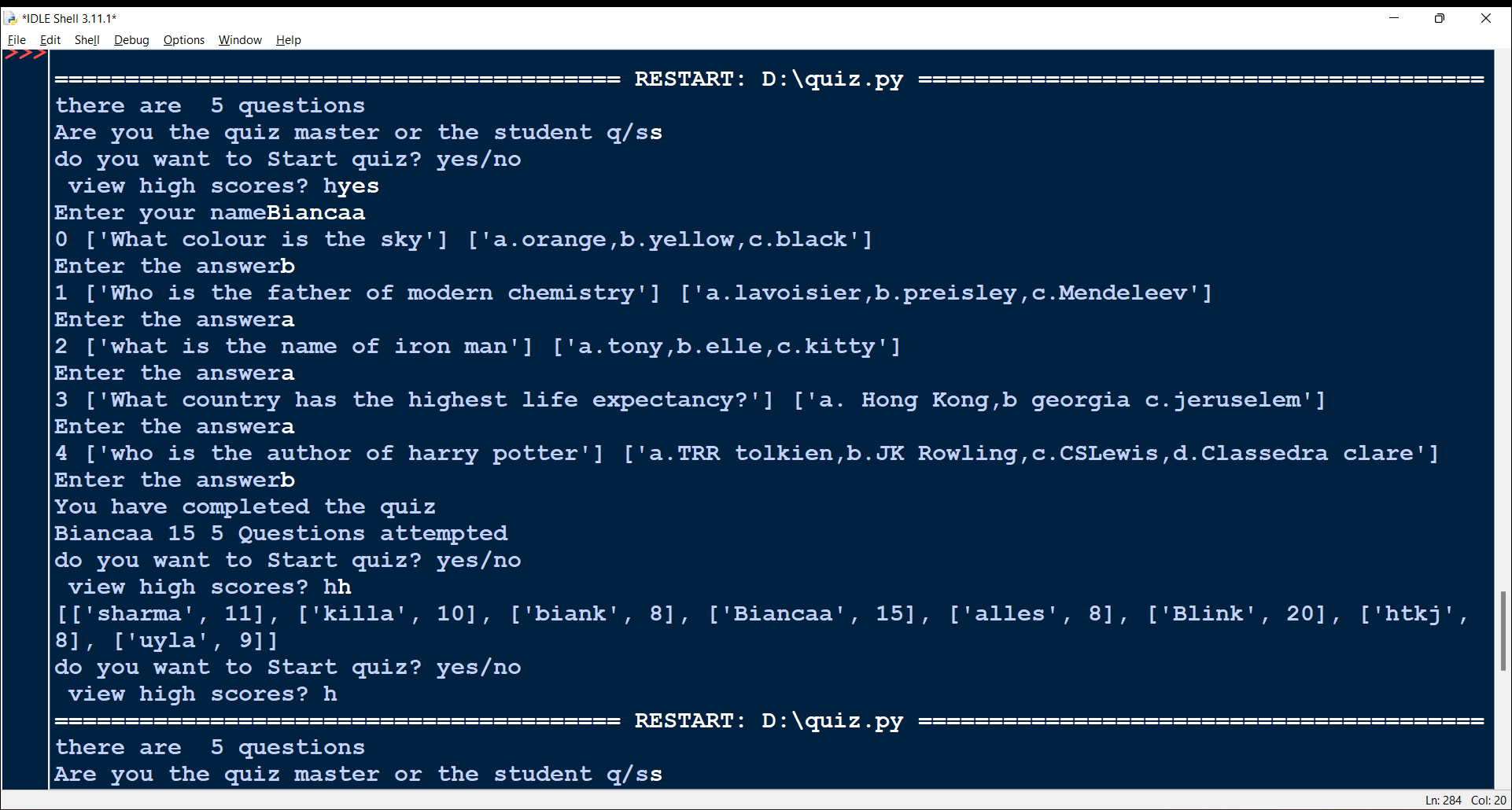
**Test cases:**

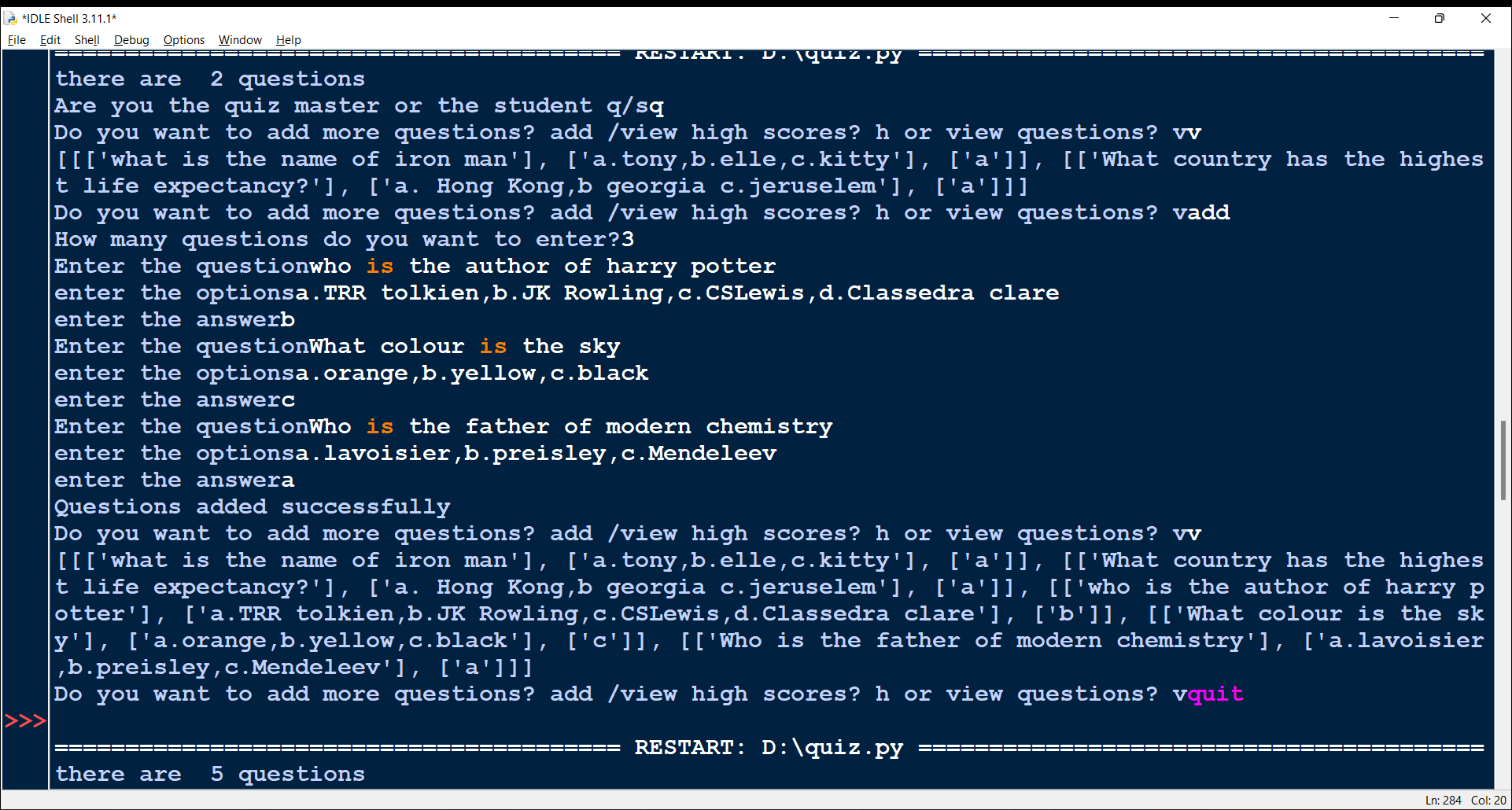
**Interface of student**

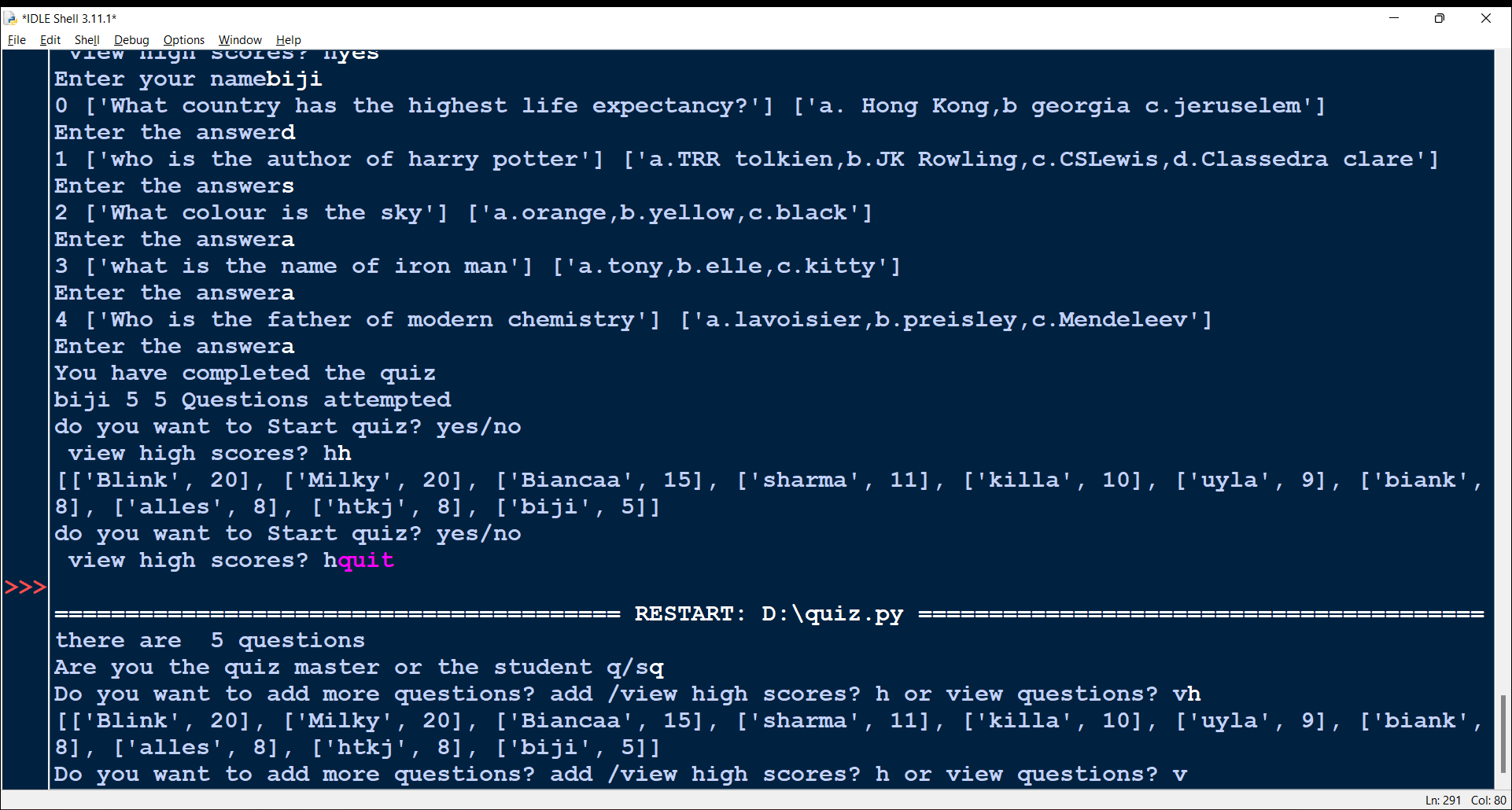
****

****

****

****

**Interface of quiz master:**

****

**Learning outcome:**

**By the end of this program I learnt how to handle multiple data and their efficient storage using files and nested lists.**