

A Project Report On

**“QUIZ”**

**Submitted By:**

Roashan Kumar

School Roll No: 30

Class: XII B

CBSE Roll No:

**Under the Guidance of**

Mr. Anoop V S

PGT (Computer Science)

Department of Computer Science

**SAINIK SCHOOL KALIKIRI**

**Department of Computer Science**

**SAINIK SCHOOL KALIKIRI**



This is to certify that **Cdt. ROASHAN KUMAR,** Roll No. 30 of Class XII has prepared the report on the Project entitled **“QUIZ”**. The report is the result of his efforts & endeavors. The report is found worthy of acceptance as final project report for the subject Computer Science of Class XII.

Signature Signature

(Internal Examiner) (External Examiner)



**DECLARATION**

I hereby declare that the project work entitled “**QUIZ**”, submitted to Department of **Computer Science**, SAINIK SCHOOL KALIKIRI is **prepared** **by me**. All the **coding** is the result of my **personal efforts**.

Cdt. Roashan Kumar

Roll No: 30

Class: XII B

SAINIK SCHOOL KALIKIRI



**ACKNOWLEDGEMENT**

I would like to express a deep sense of thanks & gratitude to my **project guide Mr. Anoop V S** Sir for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice & **constant motivation** have been responsible for the **successful** completion of this project.

My sincere thanks go to **Lt Col Susheel Kumar Mahapatro SM ,** our **Offg** **Principal** sir, for his co-ordination in extending every **possible support** for the completion of this project.

I also thanks to my **parents** for their **motivation & support**. I must thanks to my **classmates** for their timely help & support for **compilation** of this **project**.

**Last but not the least; I would like to thank all those who had helped directly or indirectly towards the completion of this project.**

Cdt. Roashan Kumar

Roll No: 30

Class: XII B

SAINIK SCHOOL KALIKIRI

**CONTENTS**

1. Working Description .........................
2. Code of the Project ............................
3. Output Screens …..............................
4. Bibliography .....................................
   * + 1. **WORKING DESCRIPTION**

The program has two main functions.

1. **TO OPEN AS ADMIN.**
   1. Here we would update and delete the questions while we work as admin.
2. **THE QUIZ GAME.**
   1. This option consists of two rounds with an assessment round to go to the last round that is the second round.
   2. The two rounds are:
      1. Rebuttal round

It consists of questions where the system asks 5

Questions and user would also ask 5 questions.

* + 1. Assessment round

It consists of a single question to go to the next round.

* + 1. Riddle round

It consist 5 questions and user has to answer them.

**2. Code of the Project**

**Note:** There 2 python files in this project

**main.py**

import mysql.connector as sq

import random as r

from mysql.connector import Error as sqlerror

from os import system

from time import sleep

from getpass import getpass

from admin import main\_admin\_menu

user\_points = 0

computer\_points = 0

random\_list = []

x = 1

def start\_program():

"""The start of the program"""

while True:

system('cls')

print("\n~~~~~~~~~~~~~ QUIZ INTERFACE ~~~~~~~~~~~~~~~")

print("Press (0) to Log in as Admin")

print("Press (1) to Play the Game")

print("Press (2) to exit the program")

user\_input = input("Please Enter your choice : ")

if user\_input == '0':

login\_as\_admin()

elif user\_input == '1':

start\_rebuttal()

else:

exit\_program()

def exit\_program():

print("Thank You for using the program.")

exit()

def login\_as\_admin():

"""This function handles the logging in of admin to the program"""

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

username\_input = input('Please Enter your username : ')

cursor.execute(f"select \* from admin\_user where username = '{username\_input}'")

data = cursor.fetchone()

if data == ():

print("The entered username does not exists in the database. Please contact the developer")

sleep(1)

start\_program()

else:

password\_input = getpass('Please Enter your password : ')

if password\_input == data[2]:

print('You have successfully logged in as admin')

sleep(1.5)

main\_admin\_menu()

else:

print(f"The entered password for the username <{username\_input}> is invalid. Please try Again")

sleep(1.5)

start\_program()

def get\_questions():

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

cursor.execute("SELECT \* FROM questions")

data = cursor.fetchall()

conn.close()

return data

def append\_mistakes(question, given\_answer, correct\_answer):

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

cursor.execute(

f"INSERT INTO mistakes VALUES ('{question}','{given\_answer}','{correct\_answer}')")

conn.commit()

conn.close()

print(f"The question has been updated successfully. Thank you")

def start\_rebuttal():

system('cls')

print(f"\n>>>>> ROUND 1 : REBUTALL QUIZ <<<<<\n")

print(f"Rules :")

print(f"1. This ROUND will consist of 10 questions")

print(f"2. User and computer can ask questions one after another. Who get's the higest points win's the game")

print(f"3. You should not ask the same question twice")

print(f"4. If the user qualifies ROUND 1. There will be a assessment question to qualify for the FINAL ROUND")

print(f"5. If you're score is less than the computer, you are not eligible for assessment")

global x, user\_points, computer\_points, username

username = input("\nEnter your name to start the game: ")

if username == '' or username == '\n':

print(f"You have not entered your name. You are not eligible to play this game")

print("Please start the game again")

exit()

while x <= 10:

print(f"Question {x}: ")

if x % 2 != 0:

computer\_question()

else:

user\_question()

x += 1

else:

print(f"### ROUND 1 COMPLETED ###")

show\_points(user\_points, computer\_points)

if user\_points > computer\_points:

start\_assessment()

else:

print(f"\nYou did not qualify the REBUTTAL ROUND. Better Luck Next Time")

exit()

def start\_assessment():

print("The Riddle to get to the Final Round")

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

cursor.execute(f"SELECT \* FROM assessment")

data = cursor.fetchone()

conn.close()

print(f"\n\*\*\* ASSESSMENT QUESTION \*\*\*")

print(f"\nQueston : {data[0]}")

user\_answer\_input = input("Enter your answer : ")

if user\_answer\_input.lower() == data[1].lower():

print(f"\n#### CORRECT ANSWER ####")

print("\nYou are qualified for the 'FINAL ROUND'")

start\_riddle()

else:

print(f"You didn't qualify the assessment. Better Luck Next Time")

exit()

def start\_riddle():

print(f"\n>>>>> FINAL ROUND : Riddle Time <<<<<\n")

print(f"Rules : ")

print(f"1. This round will consit of 5 questions")

print(f"2. Only the computer can ask questions and the user has to answer to the question")

r\_list = []

user\_score = 0

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

cursor.execute(f"SELECT \* FROM riddles")

data = cursor.fetchall()

conn.close()

for i in range(5):

random\_number = r.randint(1, len(data) - 1)

if random\_number not in r\_list:

r\_list.append(random\_number)

print(f"Riddle {i+1}")

print(f"{data[random\_number][0]}")

user\_answer = input("Enter you answer : ")

if user\_answer.lower() == data[random\_number][1].lower():

print("\nCORRECT ANSWER")

user\_score += 1

print(f"{username} Score is {user\_score}")

else:

print(f"The Correct Answer is {data[random\_number][1]}")

print(f"{username} Score is {user\_score}")

else:

continue

print(f"You have completed the FINAL ROUND")

final\_score = user\_points + 1 + user\_score

print(f"Your final score is {final\_score}")

print(f"Thank you for joining the Program")

print(f"Have a nice day!!")

exit()

def show\_points(user, comp):

global username

print(f"\n{username} obtained {user} points")

print(f"Computer obtained {comp} points\n")

def computer\_question():

global user\_points, computer\_points, random\_list

data = get\_questions()

while True:

random\_number = r.randint(0, len(data) - 1)

if random\_number not in random\_list:

random\_list.append(random\_number)

question = data[random\_number][0]

answer = data[random\_number][1]

print(f"\nComputer Turn to ask question\n")

print(f"\nQuestion : {question}")

user\_answer = input("\nEnter the answer for the question : ")

if user\_answer.lower() == answer.lower():

print(f"\nCorrect Answer")

user\_points += 1

show\_points(user\_points, computer\_points)

return None

else:

computer\_points += 1

print(f"\nWrong Answer")

print(f"The Correct Answer is {answer}")

user\_conf = input("Is the given answer correct (Y / N) : ")

if user\_conf in ['y', 'Y']:

print(f"Thank you for your confirmation.")

else:

print(f"You said that the given answer is wrong")

correct\_input = input("Please Enter the correct answer : ")

append\_mistakes(question, answer, correct\_input)

show\_points(user\_points, computer\_points)

return None

else:

continue

def user\_question():

global username, user\_points, computer\_points, random\_list

print(f"{username} Turn to ask question\n")

question\_input = input("Enter your question : ")

answer = get\_user\_answer(question\_input)

if answer:

print(f"The answer to the question is {answer}")

computer\_points += 1

else:

print(f"Couldn't find the answer to the question")

user\_points += 1

answer\_input = input(

f"Enter the correct answer for the question asked : ")

add\_question(question\_input, answer\_input)

show\_points(user\_points, computer\_points)

def get\_user\_answer(user\_question):

data = get\_questions()

for question, answer in data:

if question.lower() == user\_question.lower():

return answer

else:

return None

def add\_question(question, answer):

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

if question == '' or question == '\n':

pass

else:

try:

cursor.execute(

f"INSERT INTO questions VALUES ('{question}','{answer}')")

conn.commit()

print(f"\nThank you for the new question. I got some knowledge")

conn.close()

except sqlerror:

print(f"\nAn error occurred couldn't add the question to the database")

conn.close()

if \_\_name\_\_ == '\_\_main\_\_':

start\_program()

**admin.py**

import mysql.connector as sq

import random as r

from prettytable import PrettyTable

from mysql.connector import Error as sqlerror

from os import system

from time import sleep

from getpass import getpass

def print\_mistakes\_table():

"""This function prints the mistakes table"""

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

cursor.execute("select \* from mistakes")

data = cursor.fetchall()

table = PrettyTable()

table.field\_names = ['Question', 'Given Answer','User Given Answer']

for row in data:

table.add\_row(row)

conn.close()

return table

def return\_questions\_data():

"""Returns the prettytable of questions along with questions in dictionary format"""

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

cursor.execute("select \* from questions")

data = cursor.fetchall()

table = PrettyTable()

table.field\_names = ['Question', 'Answer']

questions = {}

for q,a in data:

table.add\_row([q,a])

questions[q] = a

conn.close()

return table, questions

def correct\_append\_mistakes():

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

print("-- Mistakes table --")

mistakes\_table = print\_mistakes\_table()

print(mistakes\_table)

print('\n')

sleep(1.5)

print("-- Questions table --")

table, questions\_dict = return\_questions\_data()

print(table)

question\_input = input("Enter the question to update it : ").lower()

if question\_input not in questions\_dict:

print("Please enter the complete question as in the table.")

return None

else:

print('Press (1) to change the question')

print('Press (2) to change the answer')

change\_input = input("Please enter you choice : ")

if change\_input not in ['1','2']:

print("You have entered an invalid choice. Please Try Again.")

elif change\_input == '1':

print("You chose to change the question.")

new\_question\_input = input("Plese Enter the new question : ")

confirm = input("Are you sure you want to update (Y / N) : ")

if confirm in ['y','Y']:

cursor.execute(f"update questions set question = '{new\_question\_input}' where question = '{question\_input}'")

conn.commit()

print("YOU HAVE SUCCESSFULLY UPDATED THE QUESTION")

else:

print("You chose not to update the question.")

return None

elif change\_input == '2':

print("You chose to change the answer for a question")

new\_answer\_input = input("Please Enter the new answer : ")

confirm = input("Are you sure you want to update (Y / N) : ")

if confirm in ['y','Y']:

cursor.execute(f"update questions set answer = '{new\_answer\_input}' where question = '{question\_input}'")

conn.commit()

print("YOU HAVE SUCCESSFULLY UPDATED THE ANSWER FOR THE QUESTION")

else:

print("You chose not to update the answer of the questions.")

return None

else:

pass

def delete\_question():

"""This function allows the admin user to delete the question from the questions table"""

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

table, questions = return\_questions\_data()

print(table)

sleep(1.5)

question\_input = input("Enter the question to delete : ")

if question\_input in questions:

confirm\_input = input("Are you sure you want to delete the question (Y / N) : ")

if confirm\_input.lower() == 'y':

cursor.execute(f"delete from questions where question = '{question\_input}'")

conn.commit()

print("\nThe question has been deleted successfully.")

conn.close()

else:

print("You chose not to delete the question.")

conn.close()

return None

else:

print("The question was not found in the database. Please enter the question as in the table")

conn.close()

return None

def clear\_mistakes\_table():

"""This function lets the admin user clears the mistakes table"""

conn = sq.connect(host='localhost', user='root',

password='student', database='quiz')

cursor = conn.cursor()

table = print\_mistakes\_table()

print(table)

sleep(1.5)

confirm\_input = input("Are you sure you want to delete the mistakes table (Y / N) : ")

if confirm\_input.lower() == 'y':

cursor.execute("delete from mistakes")

conn.commit()

print("\n")

print("Mistakes table has been successfully cleared.\n")

else:

print("You chose not to clear the mistakes table")

conn.close()

return None

def exit\_program():

print("Thank You for using the program.")

exit()

def main\_admin\_menu():

"""This function prints the admin menu and takes input from the user"""

system('cls')

while True:

print("----- ADMIN MENU -----")

print("Press (1) to append the mistakes in questions of ROUND 1")

print("Press (2) to delete a question from questions table")

print("Press (3) to clear the mistakes table")

print("Press (4) to go the main menu")

print("Press (5) to exit the program")

admin\_user\_dict = {'1' : correct\_append\_mistakes,

'2' : delete\_question,

'3' : clear\_mistakes\_table,

'5' : exit\_program}

admin\_user\_input = input("Enter you choice : ")

if admin\_user\_input in admin\_user\_dict:

admin\_user\_dict[admin\_user\_input]()

elif admin\_user\_input == '4':

return None

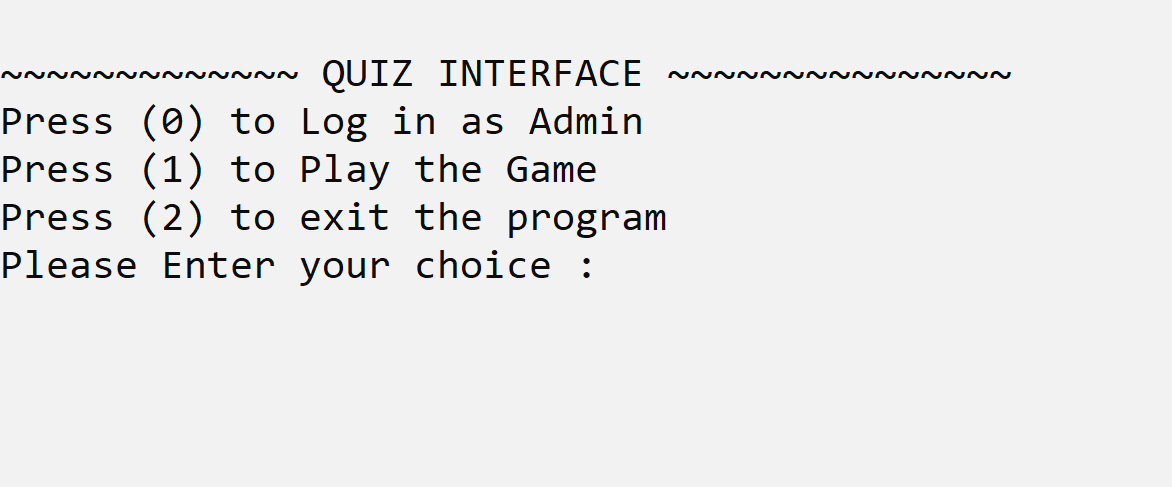
else:

print("You have entered an invalid option. Please Try Again")

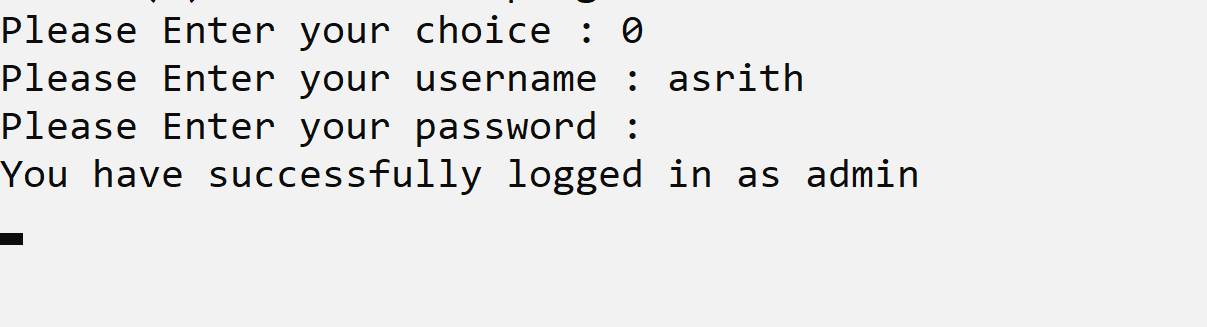
continue

**3. Output Screens**

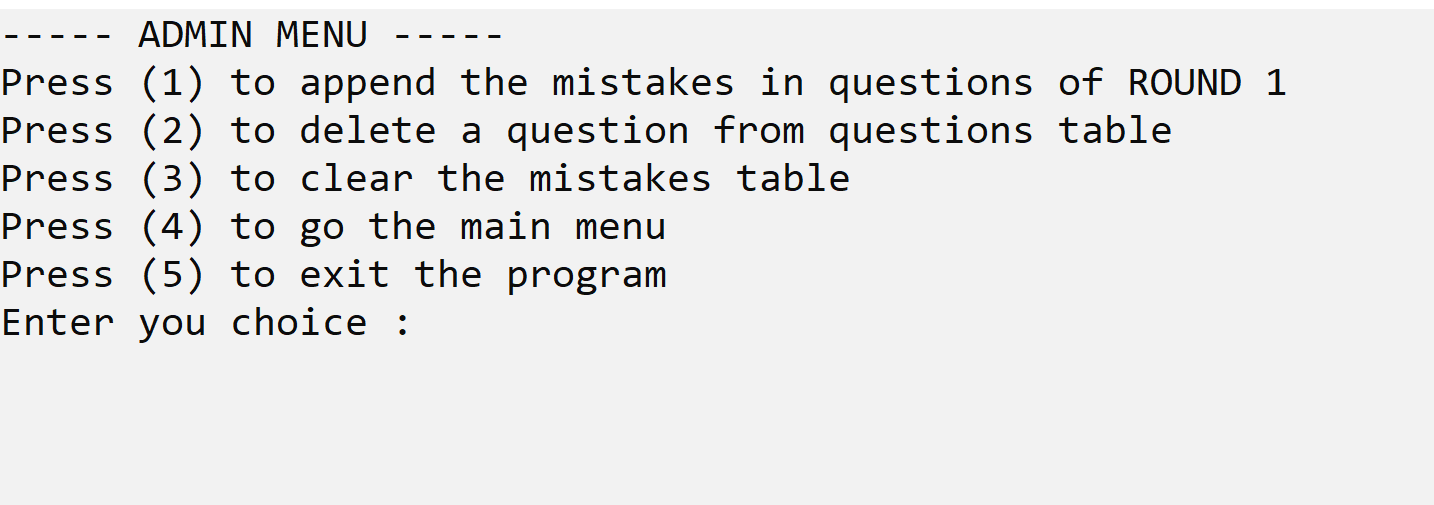
**Starting screen**



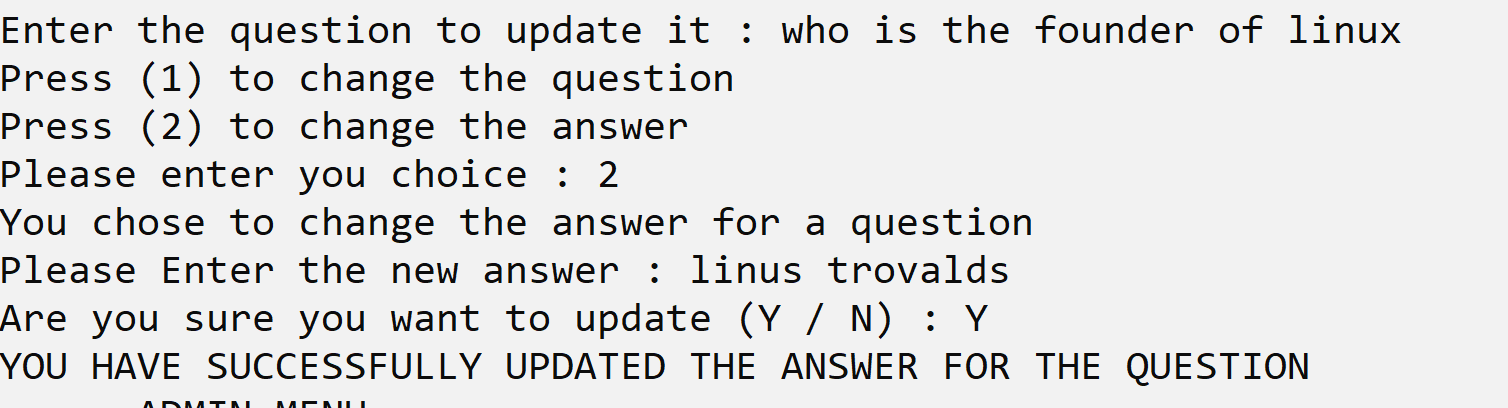
Logging in as Admin



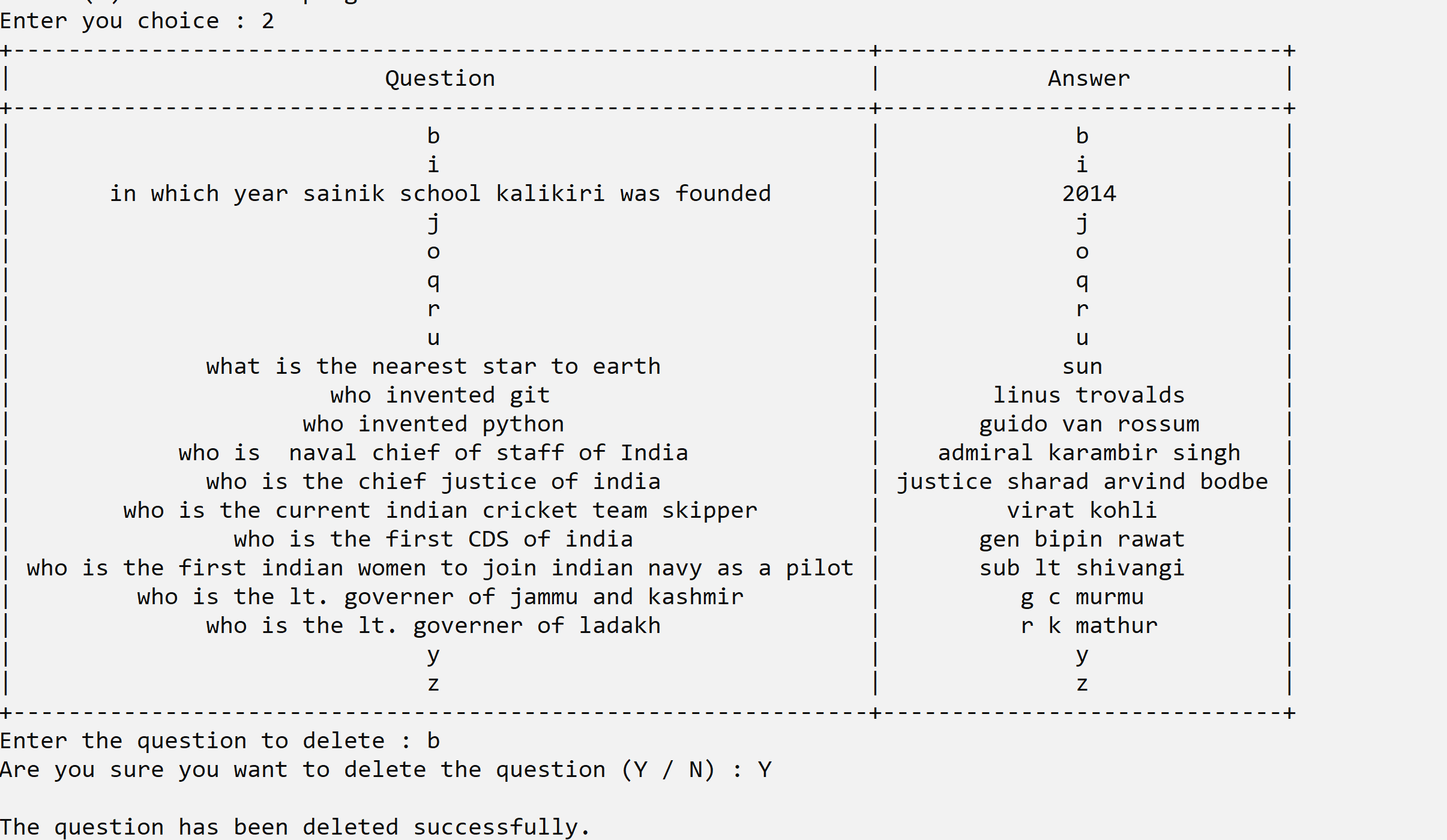
Admin Menu



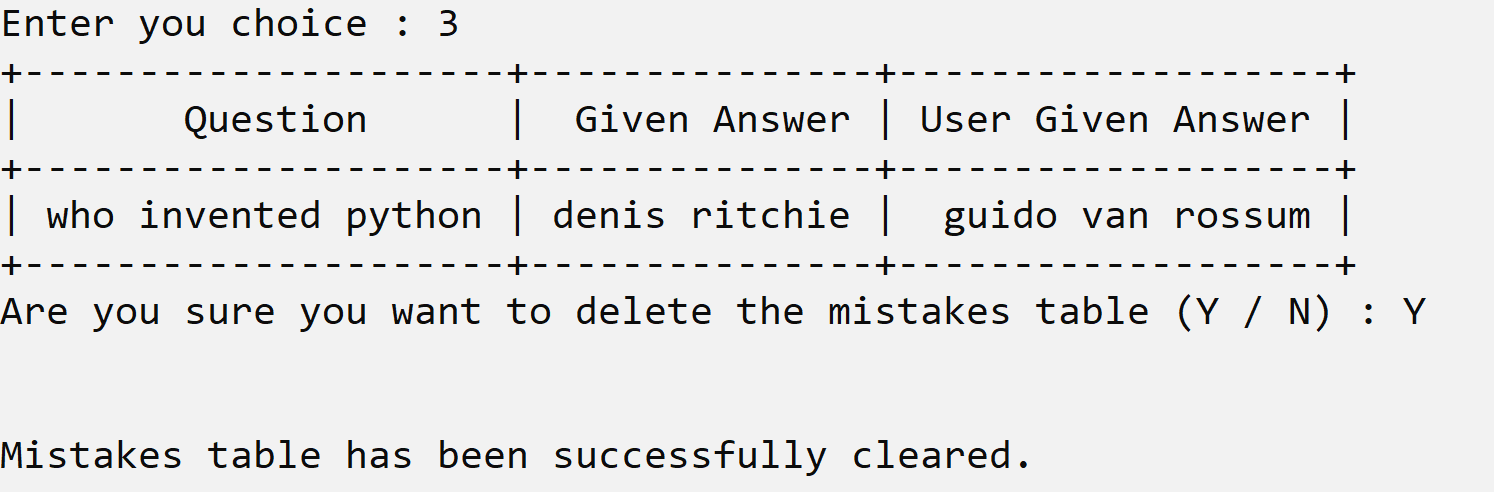
Option 1: To append the mistakes in questions of ROUND 1



Option 2: To delete a question from questions table

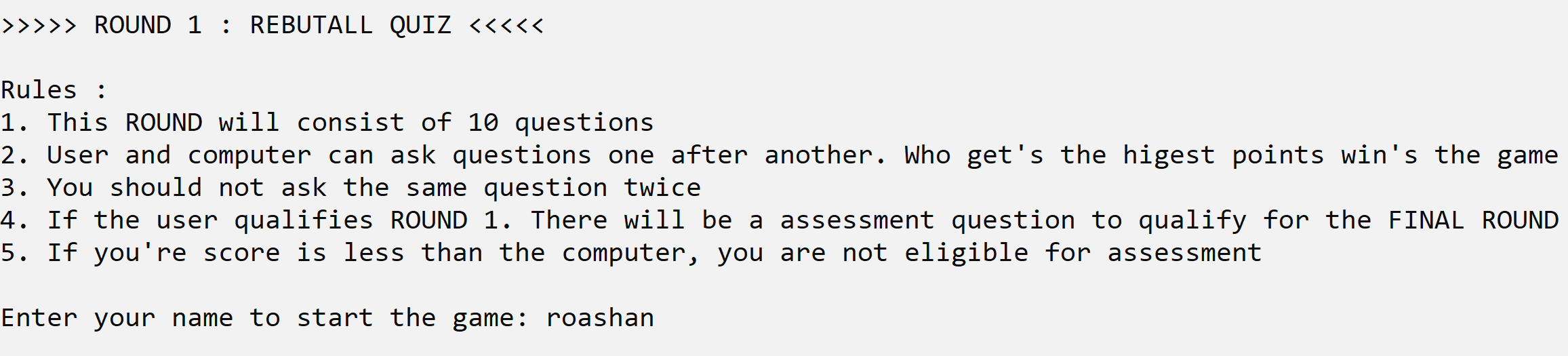


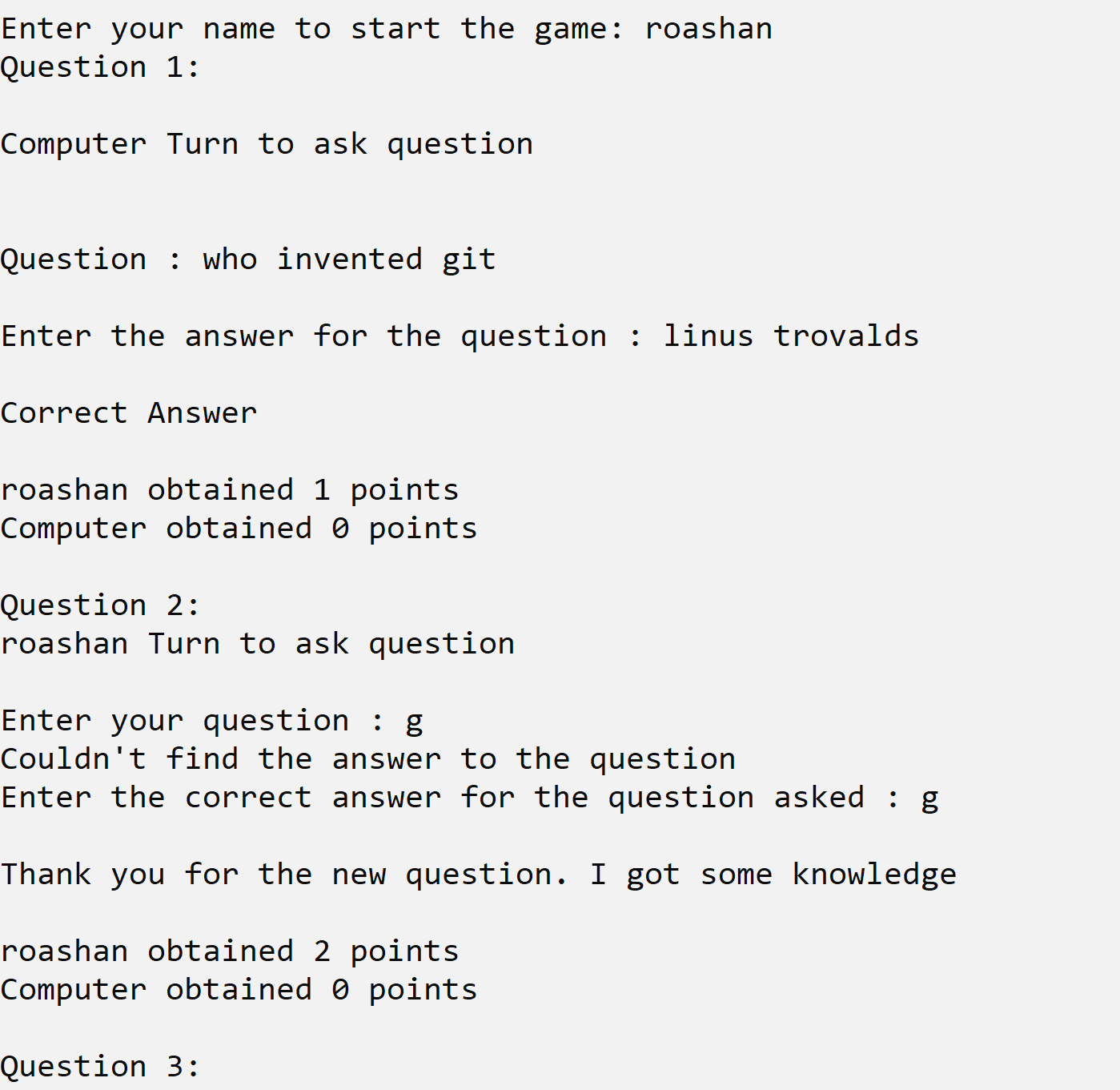
Option 3: To clear the mistakes table

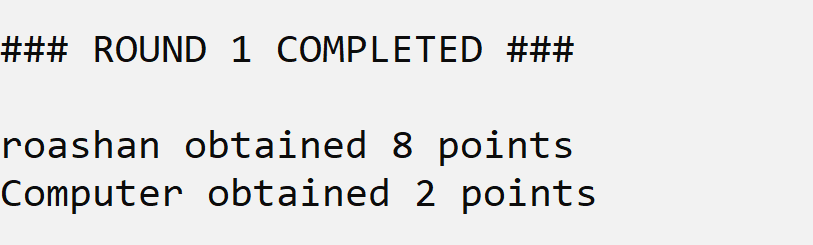


**Option 1 in Main Menu (To play the game)**

**Round one**

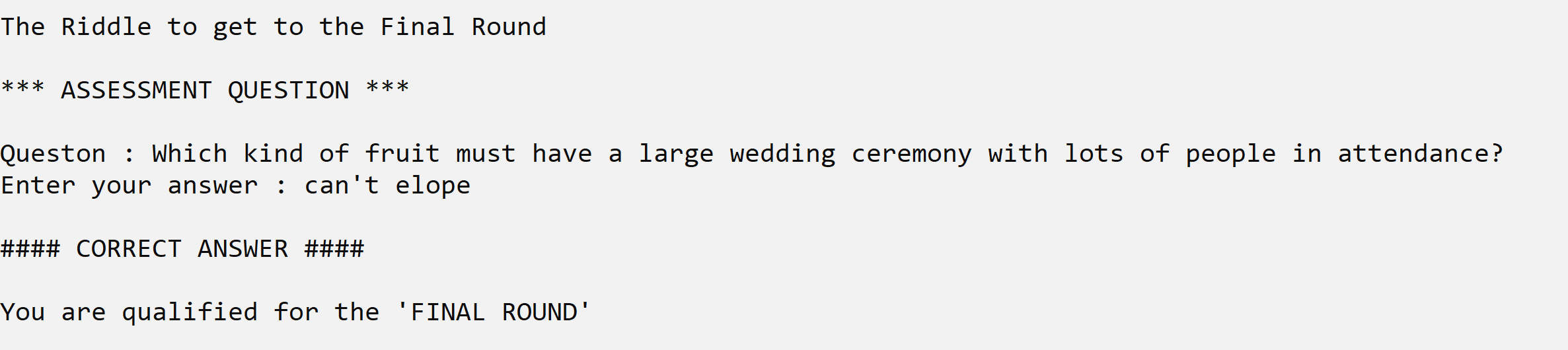






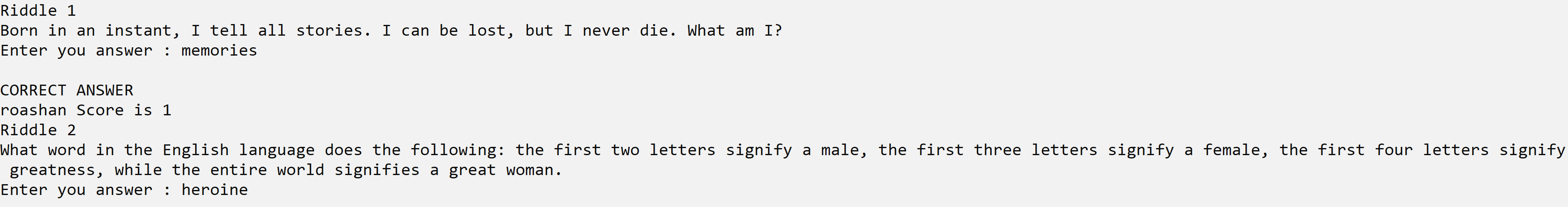
If you don’t qualify round one you can’t attend round two

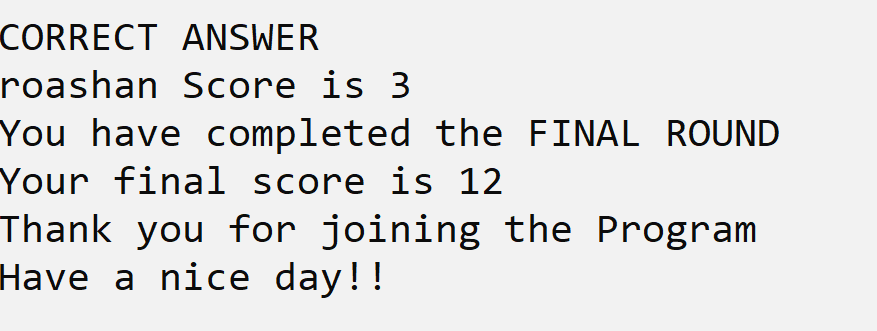
**Assessment round**



If you fail to answer the question your quiz will be completed.

**Final round(riddle round)**

****



**4. BIBLIOGRAPHY**

1. Computer Science with Python [Textbook XII] by Sumita Arora
2. https://ptable.readthedocs.io/en/latest/
3. https://docs.python.org/3/library/getpass.html