

Informatics Institute of Technology

Course: Foundation Program in Higher Education

Module Code: DOC334

Module Name: Programming II

Module Leader: Dr. Damitha Karunaratne

Assignment Type: Individual coursework

Issue Date: 10th March 2020

Hand in Date: 10th April 2020 at 10.30 am

Student Name and ID:

* Sri Harish Murugesu (2019357)

ACKNOWLEDGEMENT

In performing my coursework, I had to take the help and

guideline of our lecturers, who deserve my greatest gratitude.

I would like to show gratitude to

“Dr. Damitha Karunaratn and Mr. Nishan Hrankahawa”

for giving me a good guideline for the coursework.

Especially this itself have made valuable

comment suggestions on this proposal which gave me an

inspiration to improve my coursework. I thank all the

people for their help directly and indirectly to complete our coursework.

THE GAME

Table of Contents

[1. INTRODUCTION 5](#_Toc37278670)

[2. Quick Game 6](#_Toc37278671)

[11](#_Toc37278672)

[3. Custom Game 12](#_Toc37278673)

[4. Past player results 27](#_Toc37278674)

[5. Exit game 30](#_Toc37278675)

[6. Final program 31](#_Toc37278676)

[7. References 33](#_Toc37278677)

[Figure 1 Game menu 6](file:///C:\Users\Sri%20Harish\Desktop\Dox.docx#_Toc37276128)

[Figure 2 Quickgame 11](file:///C:\Users\Sri%20Harish\Desktop\Dox.docx#_Toc37276129)

[Figure 3 Easy mode 26](#_Toc37276130)

[Figure 4 Medium mode 26](#_Toc37276131)

[Figure 5 Hard mode 27](#_Toc37276132)

[Figure 6 Past player details 30](#_Toc37276133)

[Figure 7Exit 31](#_Toc37276134)

# 1. INTRODUCTION

A simple game that represents the users to enter the answers for the questions they get. Basically this program that I have made has basic game such as the “Quick game” and a intermediate game called the “Custom game” where the user decides whether he wants easy, medium or hard mode to test his brain skills on the math game I have made.

# Creating game menu using definitions

def Menu():

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\n Game Menu ")

print("\*1] Quick Game ")

print("\*2] Custom Game")

print("\*3] Display Past Game Results")

print("\*4] Exit")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

The Screen shot above represents a function that has been created as “Menu” defining the menu to print the options available to build the structure of the programs with the Quick game, Custom game, The past player results and the exit. We see the user should to enter the number 1], 2], 3] or 4] to enter a particular option where the option will be explained further as I come to the final program. Here is the program in action.

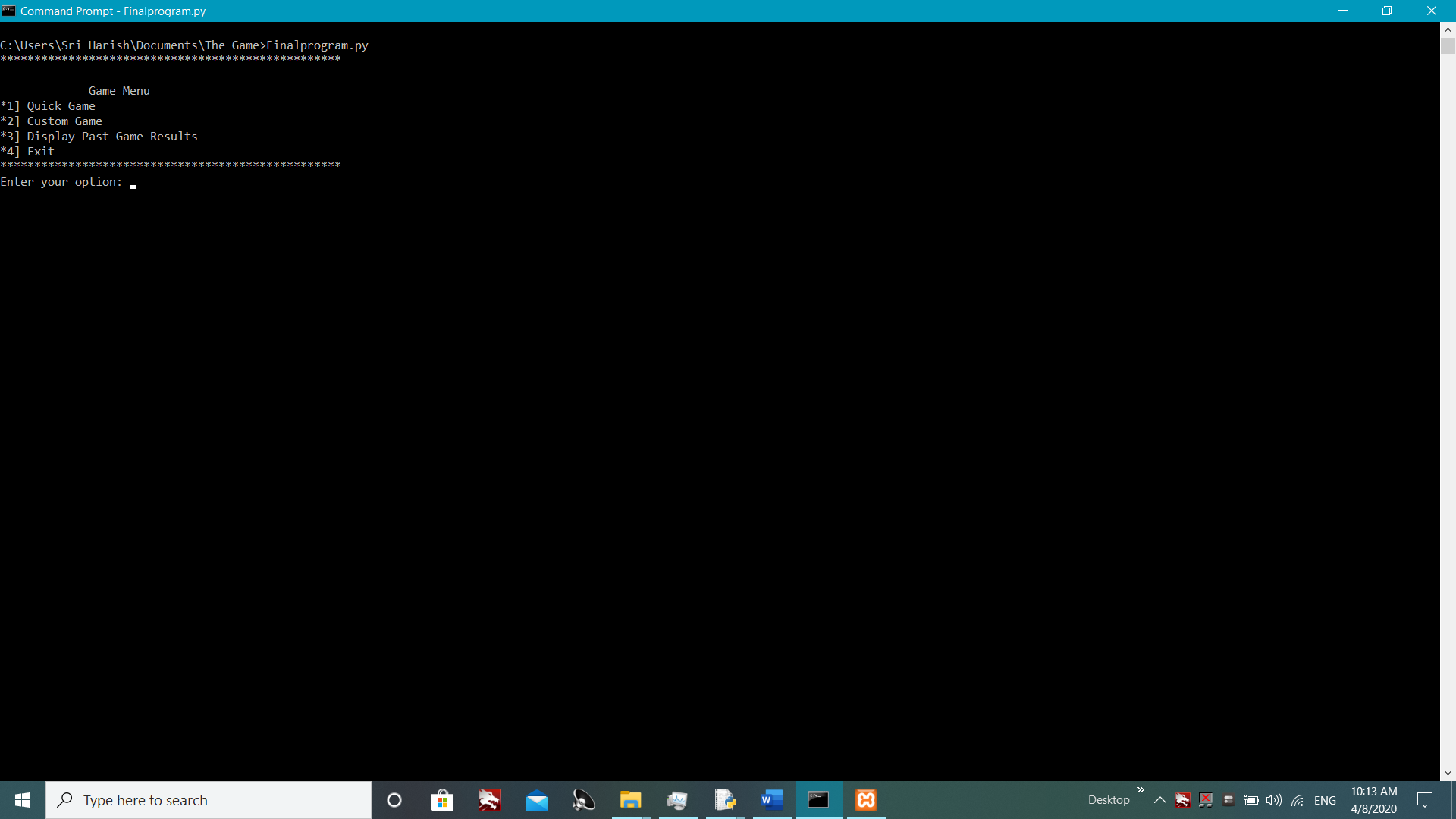


Figure Game menu

# 2. Quick Game

# Importing random in order to mix the numbers given to the user

import random

# Defining the gen number

def genNumber (maxNum):

return random.randint(0,maxNum)

# Defining and using the random to give different numbers

def Quickgame():

print("\n Quick Game ")

# Creating variables

name=input("Enter your name: ")

Co=0

In=0

num1=0

num2=0

Uans=0

ans=0

# Using for to range the numbers to maximum of 5

questions=[ ]

for i in range(5):

q = str(genNumber(10)) + " + " + str(genNumber(10))

answer = int(input(q+ " = "))

questions.append((q,answer))

# Using IF condtion to show whether the answer is correct or wrong

print("---------------------------------------------------------------------------------------------")

print("---------------------------------------------------------------------------------------------")

# Using for to give the answer and the question after typing the answer by the user

print(" Quick Results ")

for q,a in questions:

correct\_answer = eval (q)

if correct\_answer == a:

print(q,"=",a,"Correct")

Co+=1

else:

print(q,"=",a,"Incorrect", "Correct answer","[",correct\_answer,"]")

In+=1

print("------------------------------------------------------------------------------------------------")

print("------------------------------------------------------------------------------------------------")

#Programming to show the Game Results the name, the correct and wrong answers plus the final

Per=int((Co/5)\*100)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print(" GAMERESULTS ")

print("\n Your name is", name)

print("\n Questions answered is",Co)

print("\n Question not answered is",In)

print("\n Final score out of 5 is",Co)

print("\n Your percentage is", Per)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

import mysql.connector

# open database connection with a dictionery

conDict = {'host' : 'localhost', 'database' : 'dl\_game', 'user' : 'root', 'password' : ''}

db = mysql.connector.connect(\*\*conDict)

# Prepare a cursor object using cursor() method

cursor = db.cursor()

# Execute SQL query using execute() method.

mySQLText = "INSERT INTO quickgame (Name, Correct, total\_questions, Percentage) VALUES (%s, %s, 5, %s)"

val=(name, Co, Per )

cursor.execute(mySQLText,val)

db.commit()

print(cursor.rowcount, "record added")

db.close()

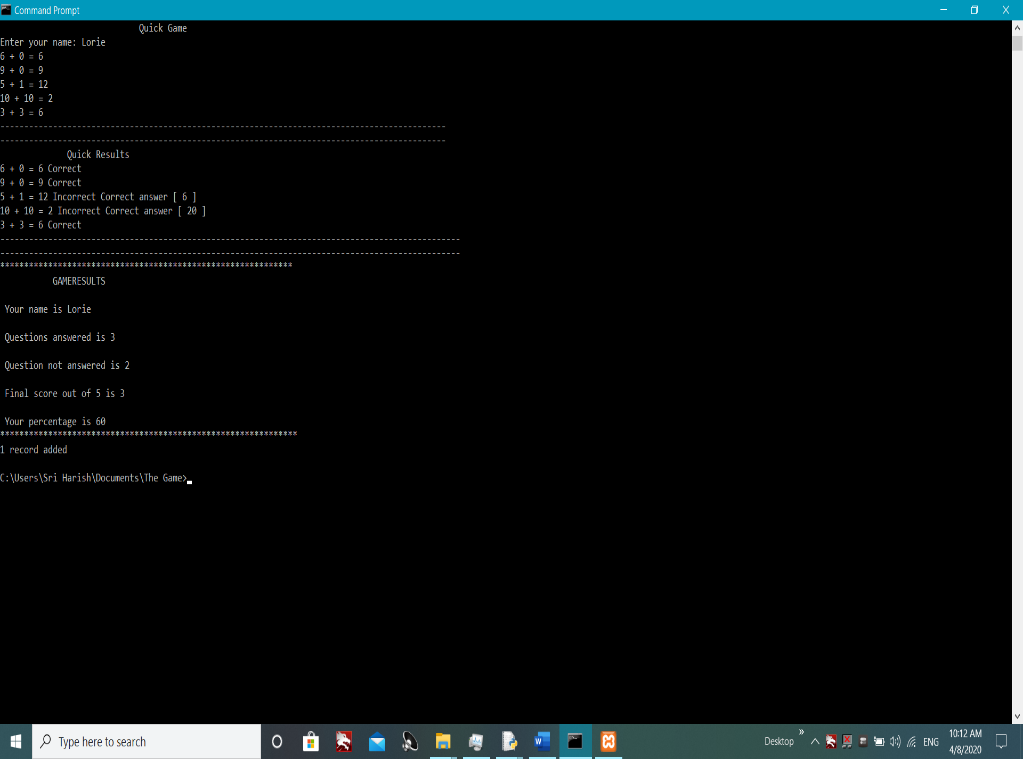
As you see the above codes this program represents the “Quick game”, where basic 5 questions will be asked by the user and the questions will only include addition based.

Importing the module random allows the program to mix up the number randomly so that the user gets different numbers and by defining a function “genNumber” this allows the program to randomize the numbers using string format. Moving on by using another function defining in “Quickgame” the program structures are typed inside. Creating variables in order to input the name of the user to make the program justify correct and wrong answers and also to record the users answers and display the two numbers used in the question “num1 and num2”.

And questions used [] is to make the list with the program to ensure that the append is able to record in the questions list. Moreover using “For range” is to make the questions set 5, where in this program the maximum questions gives is 5. And making a “q” as a variable to record and output the questions by using string method and genNumber to randomize the questions from 0 to 10 numbers. And making a another variable “answer” which the user has to enter the answer and the “questions.append” will automatically store in the list “questions = []”. More over using the “for” method record the questions and the answer to the list question. Afterwards by making a variable “correct\_answer” using “eval” to make the program accept the questions answers is correct answer.

Furthermore using “IF” condition to justify the real answer where if the user inputs correct answer (a), the program will output the result with the question (q) and (a) saying Correct by using a variable “Co” else the opposite (a) saying incorrect by using the variable “In” and revealing the real answer using the variable “correct\_answer”. And but outputting (q) and (a) again by saying the question = answer of the user and the correct answer as well.

And above this Screen shot the programs output to user that how many questions are correct and wrong with final score. Moving on by adding “Per” percentage to output the users score in the answer\* 5 / 100 to show in a database format. And for this “mysql.connector” module is imported. Where using a database connection with dictionary called “conDict” to choose the host, database, user and password by using single inverted commas ‘’. More onwards creating a variable “cursor” in order to control the records that the user enters in the program which will then record into MySQL database. The execute method MySQLText where by inserting the records that the user types into the quickgame such as the name, correct answers, total questions and the percentage and by making the users answers into strings allowing the records to record into database, similarly the “5” is being used in ‘VALUES’ is because the program of Quick game is maximum of 5 questions in total. And with the strings the “Val” variable being created to record on a proper order in order to not mix up the name and the answers and percentage in different columns in database. And finally ensuring the user that the record is being successfully recorded by showing “record added” and closing the database in order to not crash the system.



# 

Figure Quickgame

Here the below screen shot represents the quick game in action.

# 3. Custom Game

# Importing random in order to mix the numbers given to the user

import Sub.mode

# Defining the Customgame to import to main program

def Customgame():

print(" CUSTOMGAME ")

# Creating variables

mode=int(input("\n1]Easy\n2]Medium\n3]Hard\nEnter the mode: "))

# Using IF conditon to select the levels

if(mode==1):

Sub.mode.easy()

else:

if(mode==2):

Sub.mode.medium()

else:

if(mode==3):

Sub.mode.hard()

As you see the above screen shot this program represents the “custom game” where the user will be able to select three difficult modes such as easy, medium and hard to determine his mind of presence in order to make the game exciting. And by defining a function called “customgame” to import it to the final program. Creating variables made to enter the difficulty number 1, 2 or 3 by using the “IF” condition to select the mode where if the user selects 1 it is easy, 2 it is medium and 3 it is hard. In order to make the difficulties I have created a “Sub” package with a module named “mode”.

# Importing the random numbers and add multiply and subtract

import random

#Defining the gen number

def genNumber (maxNum):

return random.randint(0,maxNum)

# Using definition to make easy mode

def easy():

print(" \n Easy Mode ")

# Creating variables

name=input("Enter your name: ")

Qu=int(input("Enter the no of questions: "))

Co=0

In=0

num1=0

num2=0

Uans=0

ans=0

# Using for to range the numbers to maximum of 10

print("########################################################")

print("########################################################")

questions=[ ]

for i in range(1,Qu+1):

q = str(genNumber(10)) + " + " + str(genNumber(10))

answer = int(input(q+ " = "))

questions.append((q,answer))

# Using for to give the answer and the question after typing the answer by the user

print("#########################################################")

print("#########################################################")

for q,a in questions:

correct\_answer = eval(q)

if correct\_answer ==a:

print(q,"=",a,"Correct")

Co+=1

else:

print(q,"=",a,"Incorrect", "Correct answer","[",correct\_answer,"]")

In+=1

print("------------------------------------------------------------------------------------------------")

print("------------------------------------------------------------------------------------------------")

#Programming to show the Game Results the name, the correct and wrong answers plus the final

Per=int((Co/Qu)\*100)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print(" GAMERESULTS ")

print("\n Your name is", name)

print("\n Questions answered is",Co)

print("\n Question not answered is",In)

print("\n Final score is",Co)

print("\n Your percentage is", Per)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

import mysql.connector

# open database connection with a dictionery

conDict = {'host' : 'localhost', 'database' : 'dl\_game', 'user' : 'root', 'password' : ''}

db = mysql.connector.connect(\*\*conDict)

# Prepare a cursor object using cursor() method

cursor = db.cursor()

# Execute SQL query using execute() method.

mySQLText = "INSERT INTO customgame (Name, Correct, total\_questions, Percentage, Mode) VALUES (%s, %s, %s, %s,'Easy')"

val=(name, Co, Qu, Per)

cursor.execute(mySQLText,val)

db.commit()

print(cursor.rowcount, "record added")

db.close()

# Using definition to make medium mode

def medium():

print("\n Medium Mode ")

# Creating variables

name=input("Enter your name: ")

Qu=int(input("Enter the no of questions: "))

Co=0

In=0

num1=0

num2=0

Uans=0

ans=0

B= ["+", "-"]

# Using for to range the numbers to maximum of 50

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

questions=[ ]

for i in range(1,Qu+1):

D=random.choice(B)

q= str(genNumber(50))+ D + str(genNumber(50))

answer= int(input(q+ "="))

questions.append((q,answer))

# Using for to give the answer and the question after typing the answer by the user

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

for q,a in questions:

correct\_answer = eval(q)

if correct\_answer ==a:

print(q,"=",a,"Correct")

Co+=1

else:

print(q,"=",a,"Incorrect", "Correct answer","[",correct\_answer,"]")

In+=1

print("------------------------------------------------------------------------------------------------")

print("------------------------------------------------------------------------------------------------")

#Programming to show the Game Results the name, the correct and wrong answers plus the final

Per=int((Co/Qu)\*100)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print(" GAMERESULTS ")

print("\n Your name is", name)

print("\n Questions answered is",Co)

print("\n Question not answered is",In)

print("\n Final score is",Co)

print("\n Your percentage is", Per)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

import mysql.connector

# open database connection with a dictionery

conDict = {'host' : 'localhost', 'database' : 'dl\_game', 'user' : 'root', 'password' : ''}

db = mysql.connector.connect(\*\*conDict)

# Prepare a cursor object using cursor() method

cursor = db.cursor()

# Execute SQL query using execute() method.

mySQLText = "INSERT INTO customgame (Name, Correct, total\_questions, Percentage, Mode) VALUES (%s, %s, %s, %s,'Medium')"

val=(name, Co, Qu, Per)

cursor.execute(mySQLText,val)

db.commit()

print(cursor.rowcount, "record added")

db.close()

# Using definition to make medium mode

def hard():

print("\n Hard Mode ")

# Creating variables

name=input("Enter your name: ")

Qu=int(input("Enter the no of questions: "))

Co=0

In=0

num1=0

num2=0

Uans=0

ans=0

B= ["+", "-","\*"]

# Using for to range the numbers to maximum of 50

print("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@")

print("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@")

questions=[ ]

for i in range(1,Qu+1):

D=random.choice(B)

q= str(genNumber(50))+ D + str(genNumber(50))

answer= int(input(q+ "="))

questions.append((q,answer))

#Using for to give the answer and the question after typing the answer by the user

print("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@")

print("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@")

for q,a in questions:

correct\_answer = eval(q)

if correct\_answer ==a:

print(q,"=",a,"Correct")

Co+=1

else:

print(q,"=",a,"Incorrect", "Correct answer","[",correct\_answer,"]")

In+=1

print("------------------------------------------------------------------------------------------------")

print("------------------------------------------------------------------------------------------------")

#Programming to show the Game Results the name, the correct and wrong answers plus the final

Per=int((Co/Qu)\*100)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print(" GAMERESULTS ")

print("\n Your name is", name)

print("\n Questions answered is",Co)

print("\n Question not answered is",In)

print("\n Final score is",Co)

print("\n Your percentage is", Per)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

import mysql.connector

# open database connection with a dictionery

conDict = {'host' : 'localhost', 'database' : 'dl\_game', 'user' : 'root', 'password' : ''}

db = mysql.connector.connect(\*\*conDict)

# Prepare a cursor object using cursor() method

cursor = db.cursor()

# Execute SQL query using execute() method.

mySQLText = "INSERT INTO customgame (Name, Correct, total\_questions, Percentage, Mode) VALUES (%s, %s, %s, %s,'Hard')"

val=(name, Co, Qu, Per)

cursor.execute(mySQLText,val)

db.commit()

print(cursor.rowcount, "record added")

db.close()

Importing the module random allows the program to mix up the number randomly so that the user gets different numbers and by defining a function “genNumber” this allows the program to randomize the numbers using string format. Moving on by using another function defining in “Mode” the program structures are typed inside. Creating variables in order to input the name of the user and enter the number numbers of questions to the program justify correct and wrong answers and also to record the users answers and display the two numbers used in the question “num1 and num2”.

And questions used [] is to make the list with the program to ensure that the append is able to record in the questions list. Moreover using “For range” is to make the questions set according to the users need, where in this program the maximum questions is decided by the user. And making a “q” as a variable to record and output the questions by using string method and genNumber to randomize the questions from 0 to 10 numbers. And making a another variable “answer” which the user has to enter the answer and the “questions.append” will automatically store in the list “questions = []”. More over using the “for” method record the questions and the answer to the list question. Afterwards by making a variable “correct\_answer” using “eval” to make the program accept the questions answers is correct answer. As these go forward the mode easy is similar to the program mad Quick game, the changes that I made In medium and hard mode were by adding a variable “B” to identify the signs to be given by the program example the medium has + and -, whereas the hard has +, - and \* which will make the game even harder. And another variable made “D” to justify the signs on “B” by adding the random choice method which allow the sign randomize and give it to the program so that the program could be more interesting to the user.

Furthermore using “IF” condition to justify the real answer where if the user inputs correct answer (a), the program will output the result with the question (q) and (a) saying Correct by using a variable “Co” else the opposite (a) saying incorrect by using the variable “In” and revealing the real answer using the variable “correct\_answer”. And but outputting (q) and (a) again by saying the question = answer of the user and the correct answer as well.

And above the codes the programs output to user that how many questions are correct and wrong with final score. Moving on by adding “Per” percentage to output the users score in the answer\* 5 / 100 to show in a database format. And for this “mysql.connector” module is imported. Where using a database connection with dictionary called “conDict” to choose the host, database, user and password by using single inverted commas ‘’. More onwards creating a variable “cursor” in order to control the records that the user enters in the program which will then record into MySQL database. The execute method MySQLText where by inserting the records that the user types into the “mode” such as the name, correct answers, total questions and the percentage and by making the users answers into strings allowing the records to record into database ‘VALUES’ is because the program of “custom game” is maximum of questions selected by the user in total. And a variable created “Mode” in the MySQLText is because to show the user whether the he has chosen easy mode or medium or hard mode in the program.

And with the strings the “Val” variable being created to record on a proper order in order to not mix up the name and the answers and percentage in different columns in database. And finally ensuring the user that the record is being successfully recorded by showing “record added” and closing the database in order to not crash the system.

These are the programs I have typed in now let’s see the output screen shots of this.

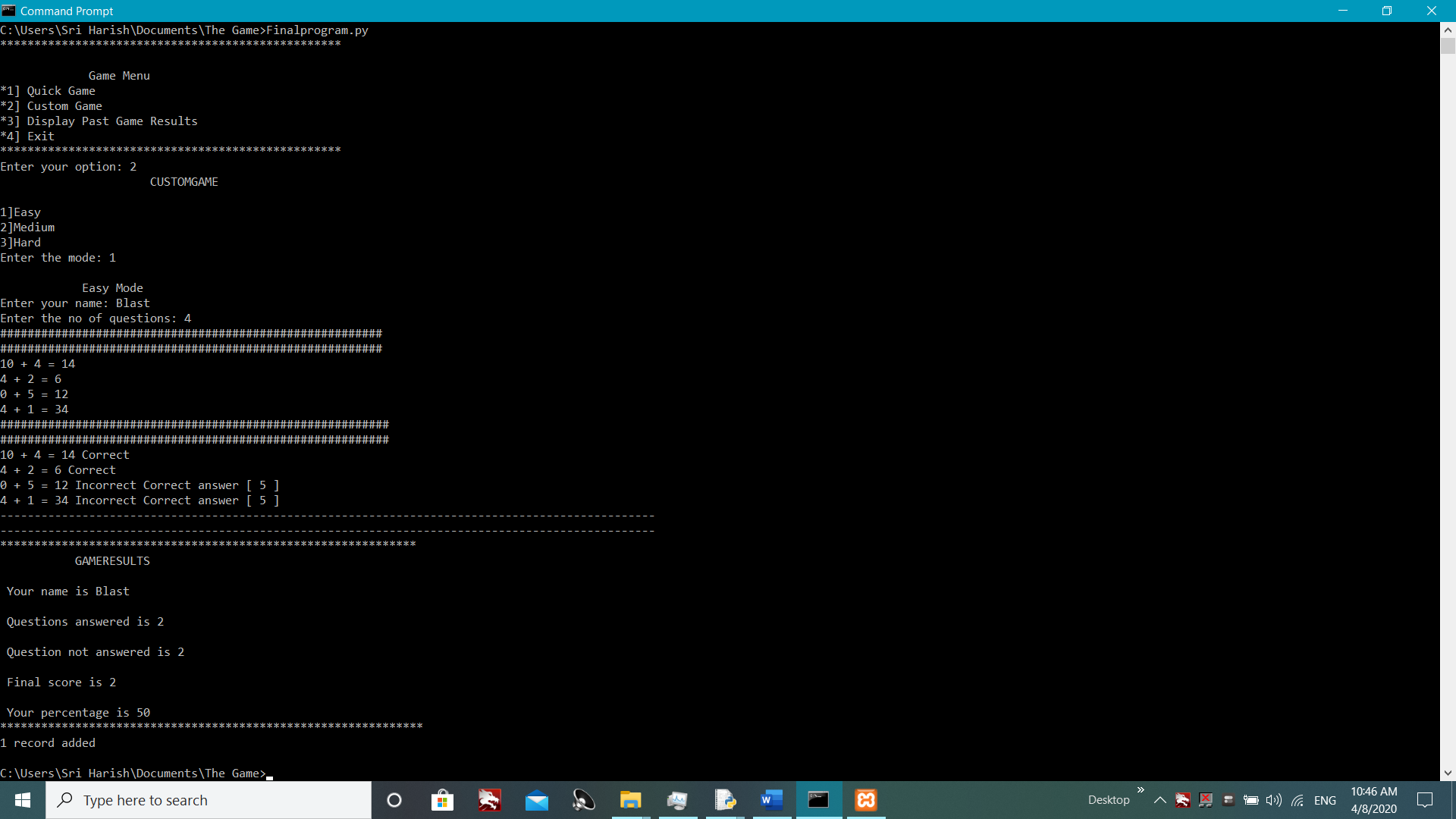


Figure Easy mode

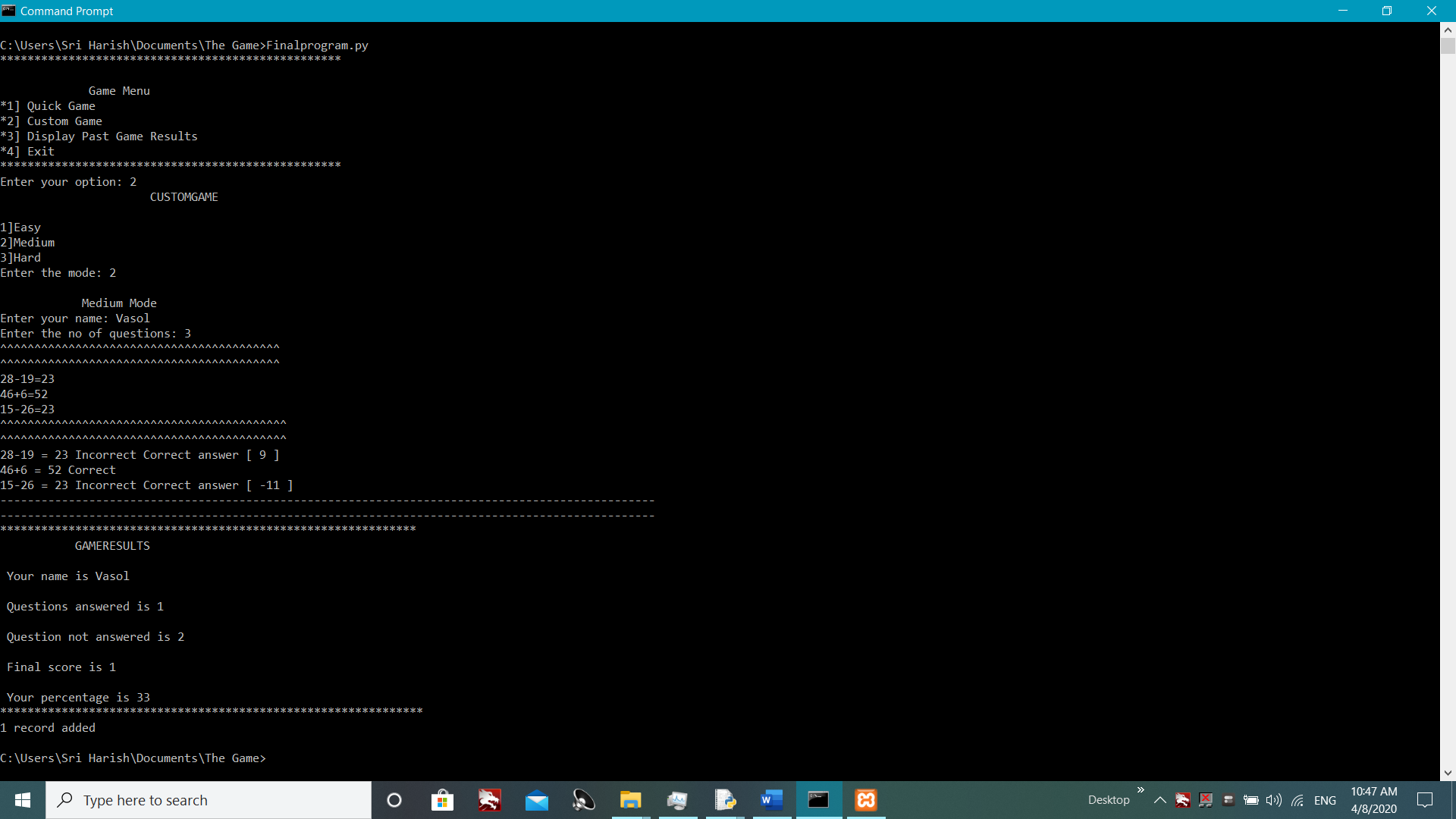


Figure Medium mode

And this is for the Hard mode below.



Figure Hard mode

# 4. Past player results

import mysql.connector

def result():

conDict = {'host':'localhost','user':'root' , 'database':"dl\_game" ,'password':''}

db=mysql.connector.connect(\*\*conDict)

# prepare a cursor object using cursor() method

cursor = db.cursor()

# execute SQL query using execute() method.

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

# Fetch results using fetchall() method.

data = cursor.fetchall()

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

print("Quickgame Results")

for item in data:

print(item)

print("[[[[[[[[[[[[[[[[[[[[[[[[[[[]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]")

print("[[[[[[[[[[[[[[[[[[[[[[[[[[[]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]")

# Disconnect from server

db.close()

conDict = {'host':'localhost','user':'root' , 'database':"dl\_game" ,'password':''}

db=mysql.connector.connect(\*\*conDict)

# prepare a cursor object using cursor() method

cursor = db.cursor()

# execute SQL query using execute() method.

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

# Fetch results using fetchall() method.

data = cursor.fetchall()

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

print("Customgame Results")

for item in data:

print(item)

print("[[[[[[[[[[[[[[[[[[[[[[[[[[[]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]")

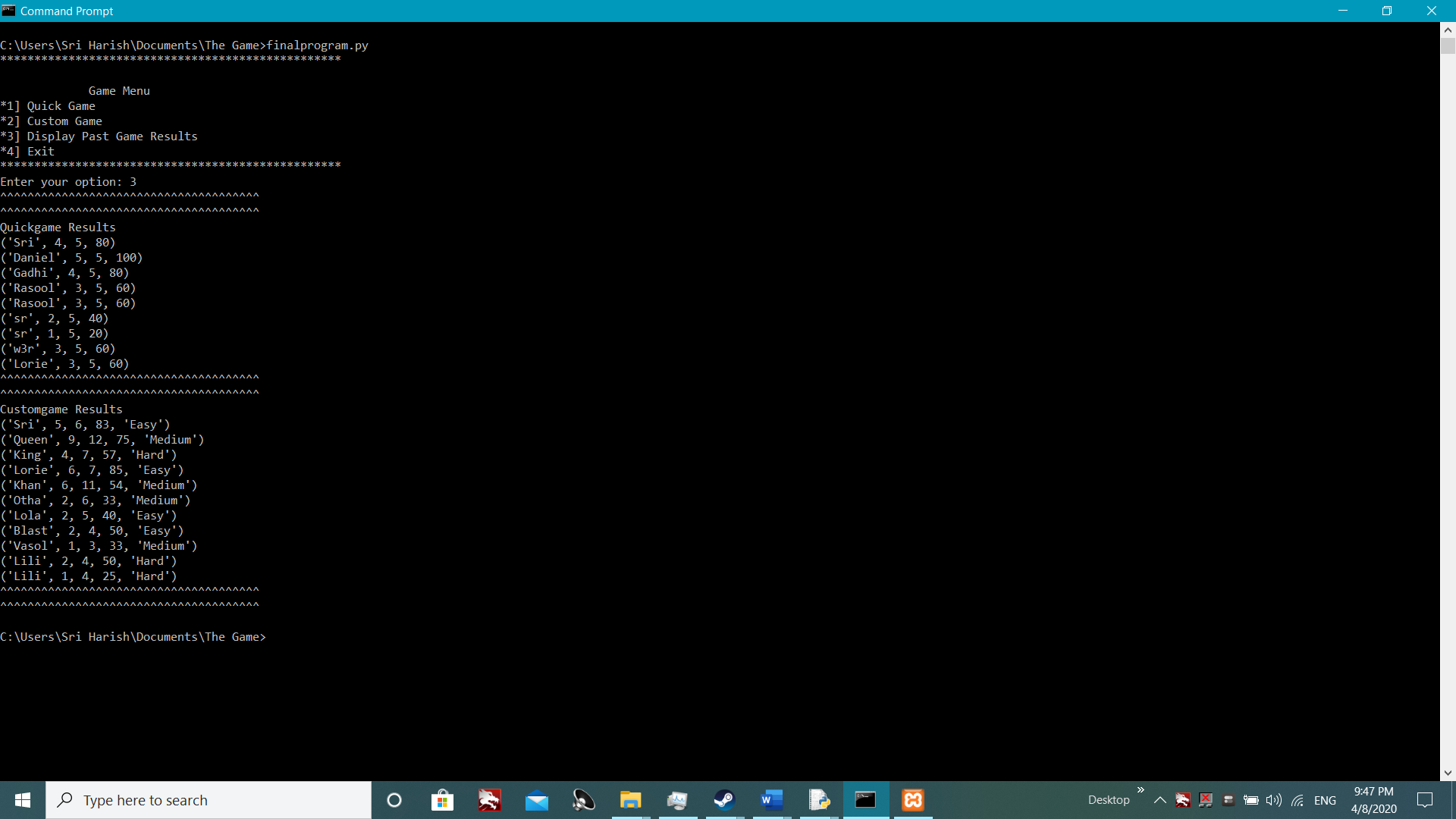
print("[[[[[[[[[[[[[[[[[[[[[[[[[[[]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]")

# Disconnect from server

db.close()

The above codes represent the program for the past player details where importing the “mysql.connector” will allow the MySQL to get and save the record in the python programs to MySQL database which will save the records entered by the user. By defining “result” as function this will then let me import it to the final program where as soon as the user enters “3” the past game results will appear as quick game results and custom game results with the use of cursor.fecthall to save the data and then by using “for” method to print all the datas that has been recorded in the past player database which is “db\_game”.

Figure Past player details



By showing the quick game results and the custom game results together.

# 5. Exit game

# Defining the exit program and closing the game

def exit():

print("Thank you for playing the game")

print("\nPress Alt+F4 to close the game :>)")

This code is a simple function that defines the “exit” in order to print the closing demonstration for the user so that the user could be impressed. Nothing but just for fancy.

Below the program is given.

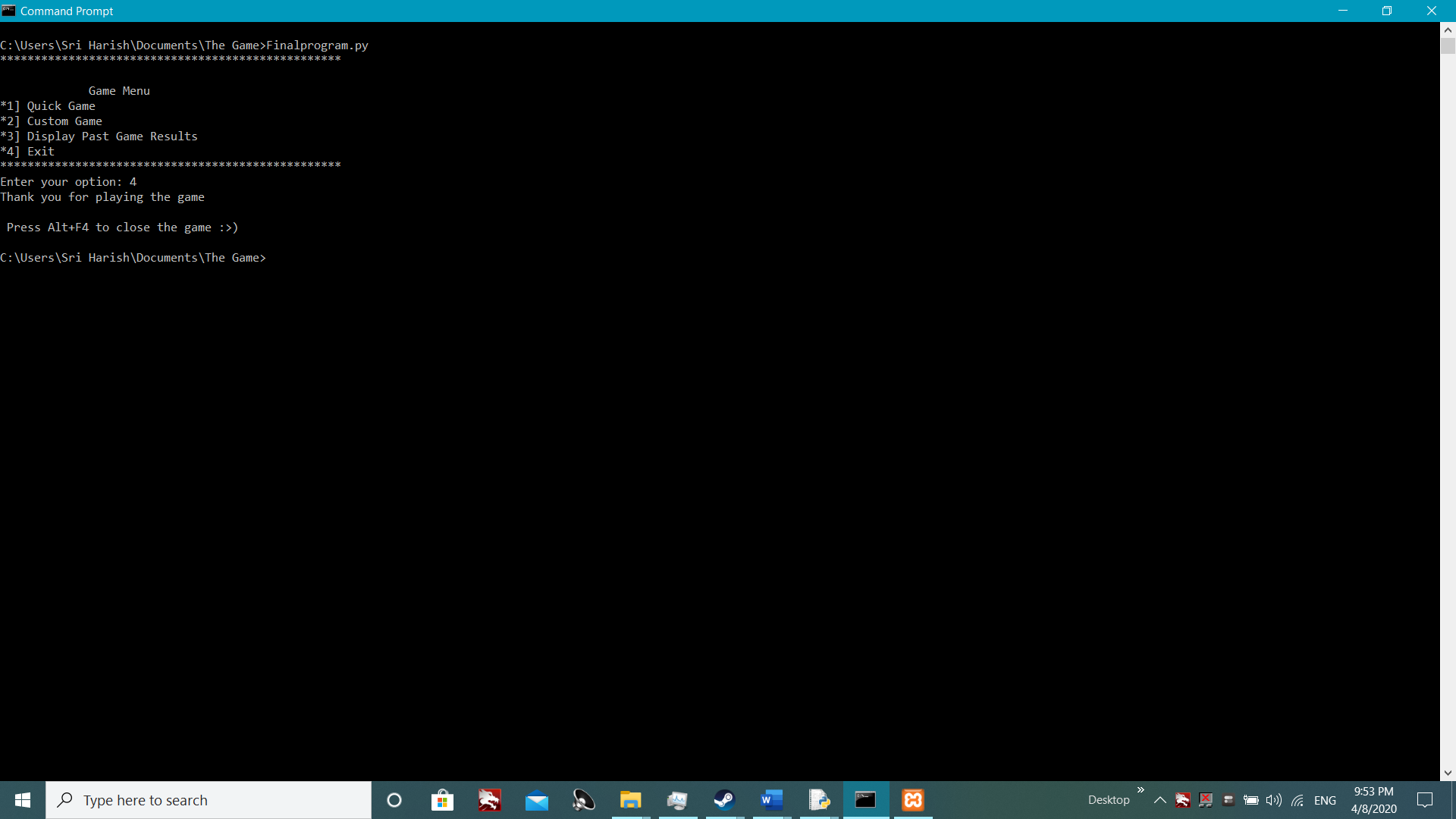


Figure Exit

# 6. Final program

# Importing all the programs to this program

import quickgame

import customgame

import pastplayer

import gamemenu

import Quitgame

# Using game menu to select available options

gamemenu.Menu()

select= int(input("Enter your option: "))

# Using IF condition to select available options

if (select==1):

quickgame.Quickgame()

else:

if(select==2):

customgame.Customgame()

else:

if(select==3):

pastplayer.result()

else:

if(select==4):

Quitgame.exit()

This is the final codes that represents the menu, the quick game, the custom game, the past player results and the exit. I have imported the modules quickgame, customgame, pastplayer and Quitgame in order to run the final program. And the gamemenu being first to show the menu first and then the option to select and the proceed to the selected options where by using “if” condition and by making “select” a variable to enter the option, where select== with determine numbers 1], 2], 3], 4] being used by the users will act according. Where 1 lead to the quick game, 2 lead to the custom game, 3 lead to the past player results and 4 lead the exit the game.

# 7. References

Lmsiit.iit.ac.lk. 2020. Informatics Institute Of Technology - LMS: Login To The Site. [online] Available at: <http://lmsiit.iit.ac.lk/lms/mod/resource/view.php?id=1456> [Accessed 8 April 2020].

R.search.yahoo.com. 2020. [online] Available at: <https://r.search.yahoo.com/\_ylt=Awr9ImJ1.41eNZEAzStXNyoA;\_ylu=X3oDMTEydGRmOGRkBGNvbG8DZ3ExBHBvcwMyBHZ0aWQDQjI5NDRfMQRzZWMDc3I-/RV=2/RE=1586392053/RO=10/RU=https%3a%2f%2fpynative.com%2fpython-random-choice%2f/RK=2/RS=zn5otJUMdckyRKtLGYa0xIpNK\_s-> [Accessed 8 April 2020].