|  |
| --- |
| **Program 01** |
| **Output** |
| ======== RESTART: /Users/biniamlemma/Desktop/CSCI\_2061/Assn\_11/new.py ========  Please enter a sentence: This is sentence 1.  Go again? y  Please enter a sentence: This is sentence 2.  Go again? y  Please enter a sentence: This is sentence 3.  Go again? y  Please enter a sentence: This is sentence 4.  Go again? n  File contents are:  This is sentence 1.  This is sentence 2.  This is sentence 3.  This is sentence 4.  Character count: 80  Alphabetic character count: 56  Numeric character count: 4  Whitespace character count: 16  Word count: 16  >>> |
| **Source Code** |
| # CSCI 2061, Assignment 10, Problem 01  # Robert Niemann  # Biniam Lemma  # This program prompts the user to enter strings and write them to a file.  # After writing it to file it open the file for input and display the contents.  # Main function  def main():  #variables  var = 'y';  charCount = 0  alphaCount = 0  numCount = 0  spaceCount = 0  wordCount = 0  # open new file called 'sentence.txt' to write the file in it  outfile = open('sentence.txt', 'w')  # prompt the user to enter the sentence and write them in the file 'sentence.txt'  while var is not 'n':  sent = input("Please enter a sentence: ")  outfile.write(sent + '\n')  var = input("Go again? ")  if var not in ('y', 'n'):  print("please for YES enter 'y' and for NO 'n'")  var = input("Go again?")    # close the file  outfile.close()  # open the file for input and diplay the content  # and count each characters, words, alphabetic caracters,  # numeric caracters, and white spaces.  infile = open('sentence.txt', 'r')  print()  print("File contents are: ")  for line in infile:  print(line, end='')  wordCount += len(line.split())  charCount += len(line)  for ch in line:  if ch.isalpha():  alphaCount += 1  elif ch.isnumeric():  numCount += 1  elif ch.isspace():  spaceCount += 1  # close the file  infile.close()  # display the values for counted words, characters, alphabetic characters,  # numeric characters, and white spaces.  print("Character count: \t\t{}".format(charCount))  print("Alphabetic character count: \t{}".format(alphaCount))  print("Numeric character count: \t{}".format(numCount))  print("Whitespace character count: \t{}".format(spaceCount))  print("Word count: \t\t\t{}".format(wordCount))      if \_\_name\_\_ == "\_\_main\_\_":  main() |

|  |
| --- |
| **Program 02** |
| **Output** |
| >>>  ======= RESTART: /Users/biniamlemma/Desktop/CSCI\_2061/Assn\_11/part2.py =======  ('Ann Annson', 19, 'Freshman', 3.0)  ('Bill Billson', 20, 'Sophmore', 3.4)  ('Carl Carlson', 21, 'Junior', 4.0)  ('Dawn Dawnson', 22, 'Senior', 2.7)  Please enter student name: Ed Edson  Please enter student age: 22  Please enter student year: Grad  Please enter student GPA: 3.9  Go again? y  Please enter student name: Fred Fredson  Please enter student age: 23  Please enter student year: Post grad  Please enter student GPA: 1.5  Go again? n  After additions, database is:  ('Ann Annson', 19, 'Freshman', 3.0)  ('Bill Billson', 20, 'Sophmore', 3.4)  ('Carl Carlson', 21, 'Junior', 4.0)  ('Dawn Dawnson', 22, 'Senior', 2.7)  ('Ed Edson', 22, 'Grad', 3.9)  ('Fred Fredson', 23, 'Post grad', 1.5)  Student to search for? Ann Annson  ('Ann Annson', 19, 'Freshman', 3.0)  Student to delete? Ed Edson  After deletion, database is:  ('Ann Annson', 19, 'Freshman', 3.0)  ('Bill Billson', 20, 'Sophmore', 3.4)  ('Carl Carlson', 21, 'Junior', 4.0)  ('Dawn Dawnson', 22, 'Senior', 2.7)  ('Fred Fredson', 23, 'Post grad', 1.5)  >>> |
| **Source Code** |
| # CSCI 2061, Assignment 10, Problem 02  # Robert Niemann  # Biniam Lemma  # This program creates a sqlite database and display. the program will  # allow the user to add more information, to search student, and to delete  # student for the database  import sqlite3  # main function  def main():  # Creating the database  studentList = sqlite3.connect('studentList.db')  studentList.execute('drop table if exists studentList')  studentList.execute('create table studentList (t1 text, i1 int, t2 text, i2 float)')  studentList.execute('insert into studentList (t1, i1, t2, i2) values (?, ?, ?, ?)', ('Ann Annson', 19, 'Freshman', 3.0))  studentList.execute('insert into studentList (t1, i1, t2, i2) values (?, ?, ?, ?)', ('Bill Billson', 20, 'Sophmore', 3.4))  studentList.execute('insert into studentList (t1, i1, t2, i2) values (?, ?, ?, ?)', ('Carl Carlson', 21, 'Junior', 4.0))  studentList.execute('insert into studentList (t1, i1, t2, i2) values (?, ?, ?, ?)', ('Dawn Dawnson', 22, 'Senior', 2.7))  # Display the database  cursor = studentList.execute('select \* from studentList order by t1')  for row in cursor:  print(row)    print()  var = 'y'  # while loop that allow the user to enter multiple student's information  while (var is not 'n'):  # prompt the to enter additional students and add those students to the database  name = input('Please enter student name: ') # enter the student's name  age = int(input('Please enter student age: ')) # enter the age  year = input('Please enter student year: ') # enter the year  GPA = float(input('Please enter student GPA: ')) # enter the GPA  # add the student into the studentList  studentList.execute('insert into studentList (t1, i1, t2, i2) values (?, ?, ?, ?)', (name, age, year, GPA))  # Ask the user if they want to enter another student's information  # If 'y' for yes go again or if 'n' for no stop.  var = input('Go again? ')  # input validation  if var not in ('y', 'n'):  print("Please enter 'y' for yes and 'n' for no ")  var = input('Go again? ')  continue    studentList.commit()  # Display the database after the additions  print()  print("After additions, database is: ")  cursor = studentList.execute('select \* from studentList order by t1')  for row in cursor:  print(row)  # Prompt the user to enter name to search from the database  print()  name = input('Student to search for? ')  cursor = studentList.execute('select \* from studentList order by t1')  studentList.commit()  # Search the student; if exist display the information  for row in cursor:  if (row[0].upper() == name.upper()):  print(row)  # Prompt the user to enter name to delete from the database  print()  name = input('Student to delete? ')    # Search the student and delete from the database  studentList.execute('delete from studentList where t1 = ?', (name,))  studentList.commit()  # display the rest data in the database  print()  print('After deletion, database is: ')  cursor = studentList.execute('select \* from studentList order by t1')  for row in cursor:  print(row)    if \_\_name\_\_ == "\_\_main\_\_":  main() |

|  |
| --- |
| **Program 03** |
| **Output** |
|  |
| **Source Code** |
|  |

|  |
| --- |
| **Program 04** |
| **Output** |
|  |
| **Source Code** |
|  |

|  |
| --- |
| **Program 05** |
| **Output** |
|  |
| **Source Code** |
|  |