|  |
| --- |
| **Program 01** |
| **Output** |
| 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: 76  Alphabetic character count: 56  Numeric character count: 4  Whitespace character count: 12  Word count: 16 |
| **Source Code** |
| #This program asks the user to type in a sentence.  #It then keeps asking them for more until  #the user says no. It then prionts out the  #number of charcters, alphabetic characters,  #white spaces characters, and the total word count.  def main():  file = open('sentences.txt', 'w')  again = 'Y'  chars = 0  alphaChars = 0  nums = 0  spaces = 0  words = 0  #boolean while loop for writing sentences to a file  while(again.lower() == 'y'):  temp = input('Please enter a sentence: ')  file.write(temp + '\n')  again = input('Go again? ')    #Closes the file that was written to  file.close()  #Opens the file created by the user  file = open('sentences.txt', 'r')  print()  print('File contents are:')  #For loops for processing characters  for line in file:  temp = line.strip()  print(temp)    for char in temp:  chars += 1  if char.isalpha():  alphaChars += 1  if char.isdigit():  nums += 1  if char.isspace():  spaces += 1  temp = line.split()  for each in temp:  words += 1  #Prints out the table in a two column table  print('Character count:\t\t\t{}'.format(chars))  print('Alphabetic character count:\t\t{}'.format(alphaChars))  print('Numeric character count:\t\t{}'.format(nums))  print('Whitespace character count:\t\t{}'.format(spaces))  print('Word count:\t\t\t\t{}'.format(words))  if \_\_name\_\_ == "\_\_main\_\_": main() |

|  |
| --- |
| **Program 02** |
| **Output** |
| After creation database is:  ('Dawn Dawnson', 22.0, 'Senior', 2.7)  ('Ann Annson', 19.0, 'Freshman', 3.0)  ('Bill Billson', 20.0, 'Sophmore', 3.4)  ('Carl Carlson', 21.0, 'Junior', 4.0)  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:  ('Fred Fredson', 23.0, 'Post grad', 1.5)  ('Dawn Dawnson', 22.0, 'Senior', 2.7)  ('Ann Annson', 19.0, 'Freshman', 3.0)  ('Bill Billson', 20.0, 'Sophmore', 3.4)  ('Ed Edson', 22.0, 'Grad', 3.9)  ('Carl Carlson', 21.0, 'Junior', 4.0)  Student to search for? Ann Annson  ('Ann Annson', 19, 'Freshman', 3.0)  Student to delete? Ed Edson  After deletion, database is:  ('Fred Fredson', 23.0, 'Post grad', 1.5)  ('Dawn Dawnson', 22.0, 'Senior', 2.7)  ('Ann Annson', 19.0, 'Freshman', 3.0)  ('Bill Billson', 20.0, 'Sophmore', 3.4)  ('Carl Carlson', 21.0, 'Junior', 4.0) |
| **Source Code** |
| #This program creates a database then adds a list in the form  #of a four column table. It then utilizes CRUD functionality to  #create, retrieve, update (not included here because of tuple use),  #and delete from the database. It prompts the user to add rows and  #lets them choose when to stop. It then reprints the database in tabular  #form and then makes the user search for a single student in the database  #and prints that individual users row. It then asks you to delete a sinlge row  #and then prints the final table.  import sqlite3  #Create portion of CRUD  def insert(db, row):  db.execute('insert into students (name, age, class, gpa) values (?, ?, ?, ?)', (row[0], row[1], row[2], row[3]))  db.commit()  #Retrieve portion of CRUD  def retrieve(db, name):  cursor = db.execute('select \* from students where name = ?', (name,))  print(cursor.fetchone())  #Delete portion of CRUD  def delete(db, name):  db.execute('delete from students where name = ?', (name,))  db.commit()  #Row-factory print rows function  def disp\_rows(db):  cursor = db.execute('select \* from students order by gpa')  for row in cursor:  print('(\'{}\', {}, \'{}\', {})'.format(row[0], float(row[1]), row[2], row[3]))  def main():  studentList = [ ('Ann Annson', 19, 'Freshman', 3.0),  ('Bill Billson', 20, 'Sophmore', 3.4),  ('Carl Carlson', 21, 'Junior', 4.0),  ('Dawn Dawnson', 22, 'Senior', 2.7)]  #Creates the students.db database  db = sqlite3.connect('students.db')  print('After creation database is: ')  db.execute('drop table if exists students')  db.execute('create table students (name TEXT PRIMARY KEY, age NEAR, class TEXT, gpa REAL)')  #Adds the full studentList  for each in studentList:  insert(db, each)  #Prints the entire database in table form  disp\_rows(db)  print()  #Boolean flag while loop for adding desired number of students  flag = 'y'  while (flag.lower() == 'y'):  name = input('Please enter student name: ')  age = input('Please enter student age: ')  year = input('Please enter student year: ')  gpa = float(input('Please enter student gpa: '))  insert(db, (name, age, year, gpa))  flag = input('Go again? ')  #Prints uopdated table from database  print('\nAfter additions, database is: ')  disp\_rows(db)  print()  #Prompts user to search database and then displays their row  search = input('Student to search for? ')  retrieve(db, search)  print()  #Prompts user to delete a student from the database  bye = input('Student to delete? ')  delete(db, bye)  #Prints uopdated table from database  print('\nAfter deletion, database is:')  disp\_rows(db)  if \_\_name\_\_ == "\_\_main\_\_": main() |

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

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

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