Unit 3 - Phonebook

Planning:

Timeline:

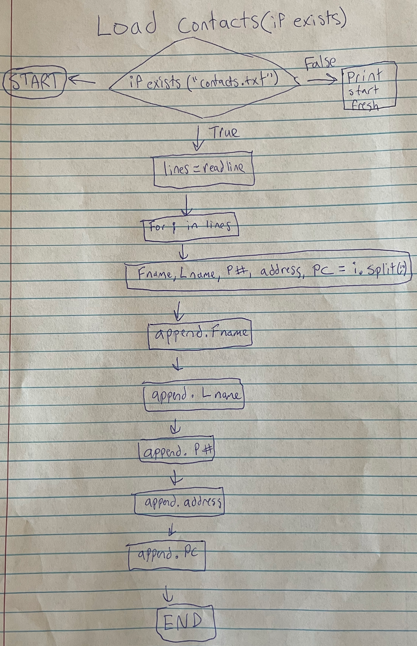
| **Date** | **Goal** |
| --- | --- |
| October 3, 2023 | Review code and do planning  Requirements: flowcharts, Timeline |
| October 4, 2023 | Program Journal  Implement code  (crash test) - if time |
| October 5, 2023 | Complete if not finished  Crash test |

Program requirements:

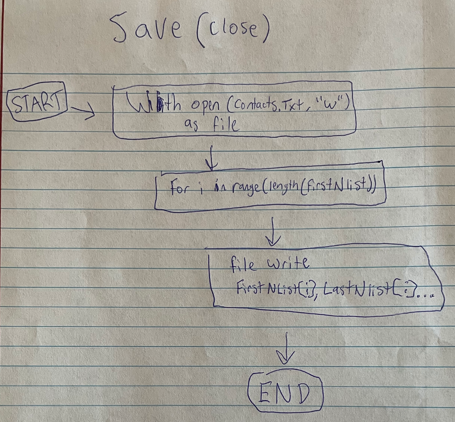
* Create useful work functions
* The user must be protected from making basic errors
* Must be uncrashable
* The program must read the contact file into the main memory
* The program must store the contact data before exiting
* The program must keep track of the changes made by the user
* Must include all the commands

October 3, 2023

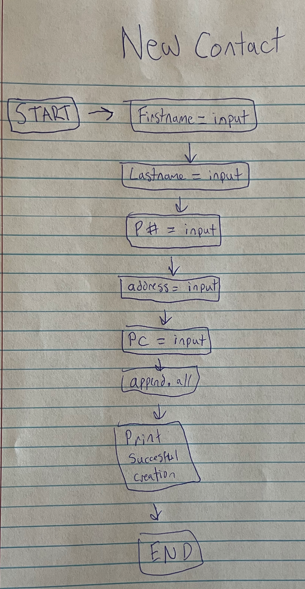
Flowcharts:[2:03] Loadcontacts():



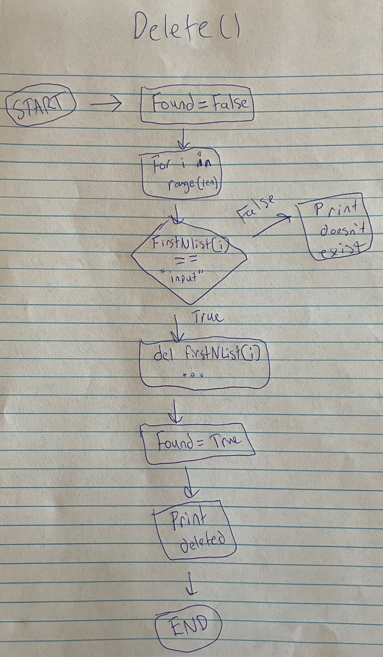
[2:20] Savefile():



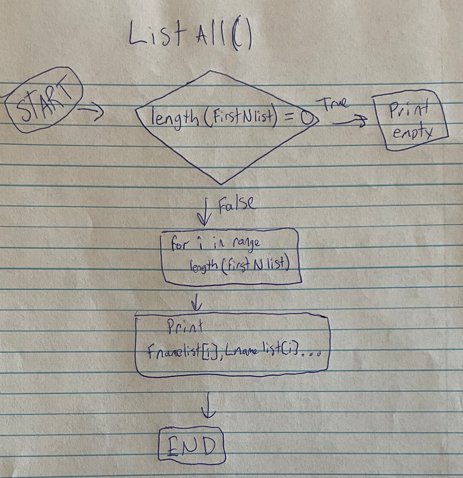
[2:25] Newcontact():



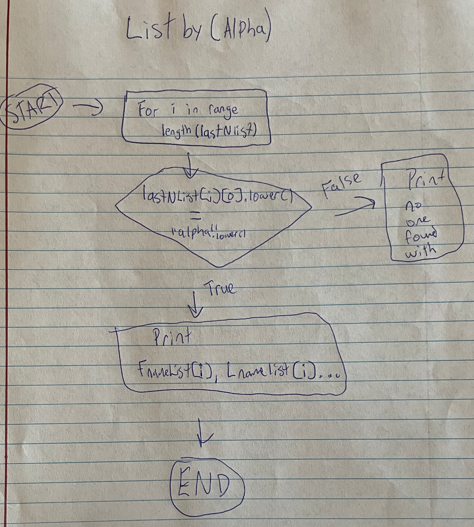
[2:40] Deletecontact():



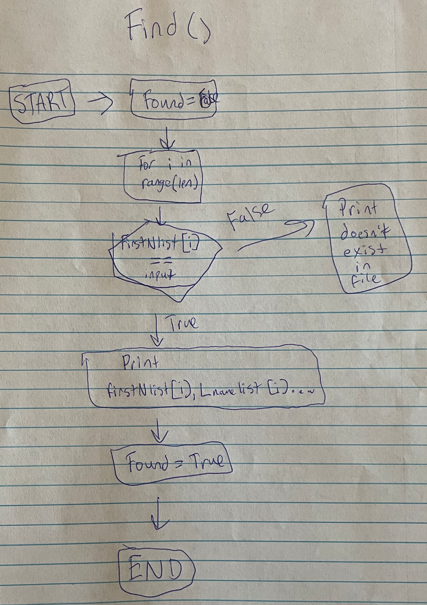
[2:50] ListAll():



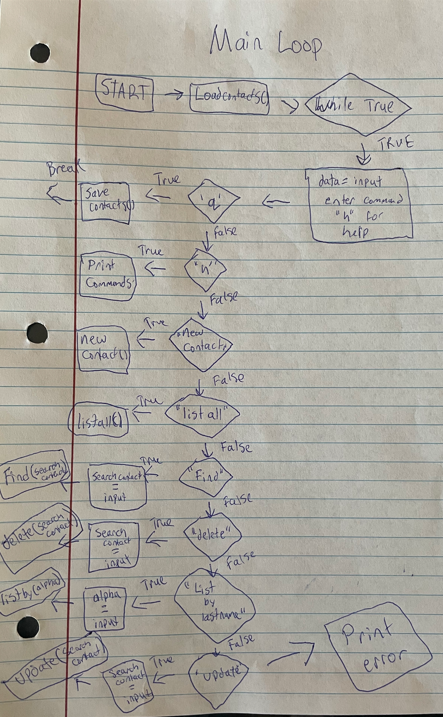
[3:00] Listbylastname(Alpha):



[3:10] Findcontact():



[3:40] main loop:



October 4, 2023

Implementing code:

[5:41] **Functions:**

At first I will start off by adding all the functions from my flowcharts into the code. I had to make a lot of changes to account for capitalizations so it wouldn’t be too harsh on the user. I was having trouble testing out my loadcontacts function as it wasnt stating that there was no file that exists. I used the Try and except expressions from chapter 17 and I moved the data input to choose what command to use further down after the load contacts function. After that this function is working as intended with my testing. I had to make some changes to the find contact function to make sure the user can enter two pieces of information to find the contact. I changed the find contact function to only account for one field because I thought it would be better because if the user only knows one thing from the contact they would never be able to find it.

# Define Your Functions Here:

# create a new contact with five basic parts

def newContact():

# input info for five basic parts

firstName = input("please enter first name: ")

lastName = input("please enter last name: ")

phoneNumber = input("please enter phone number: ")

address = input("please enter address: ")

postalCode = input("please enter postalcode: ")

# add all info to the different lists

firstNlist.append(firstName)

lastNlist.append(lastName)

phoneList.append(phoneNumber)

addressList.append(address)

postalCodeList.append(postalCode)

# let the user know it was succesfully created

print("contact succesfully added.")

# deletes a contact based on matching first and last name (assume unique)

def deleteContact(searchContact):

found = False #assume false unless find in contacts

for i in range(len(firstNlist)):

if firstNlist[i].lower() == searchContact.lower() and lastNlist[i].lower() == searchContact.lower():

del firstNlist[i]

del lastNlist[i]

del phoneList[i]

del addressList[i]

del phoneList[i]

found = True

print("contact succesfully deleted.")

else:

print("name not found in contacts...")

# takes in a first or last name, phone number or postal code

def findContact(searchContact):

found = False

for i in range(len(firstNlist)):

if firstName[i].lower() == searchContact.lower() or lastName[i].lower() == searchContact.lower() or phoneList[i] == searchContact.lower() or addressList[i] == searchContact.lower() or postalCodeList[i] == searchContact.lower():

print(f"{firstNlist[i]}, {lastNlist[i]}, {phoneList[i]}, {addressList[i]}, {postalCodeList[i]}")

found = True

if not found:

print("contact not found.")

# update contact

def updateContact(firstName, lastName):

firstName = input("Enter the first name of the contact to update: ")

lastName = input("Enter the last name of the contact to update: ")

# Search for the contact in the lists

index = -1 # start with invalid index

for i in range(len(firstNlist)):

if firstNlist[i].lower() == firstName.lower() and lastNlist[i].lower() == lastName.lower():

index = i

break

if index != -1:

# contact found ask for update

new\_firstName = input("Enter the new first name: ")

new\_lastName = input("Enter the new last name: ")

new\_phoneNumber = input("Enter the new phone number: ")

new\_address = input("Enter the new address: ")

new\_postalCode = input("Enter the new postal code: ")

# update the list

firstNlist[index] = new\_firstName

lastNlist[index] = new\_lastName

phoneList[index] = new\_phoneNumber

addressList[index] = new\_address

postalCodeList[index] = new\_postalCode

print("contact succesfully updated.")

else:

print("contact doesn't exist.")

# Continue creating useful functions below with your own criteria

# save contacts when close

def saveContacts():

with open("contacts.txt", "w") as file:

for i in range(len(firstNlist)):

file.write(f"{firstNlist[i]}, {lastNlist[i]}, {phoneList[i]}, {addressList[i]}, {postalCodeList[i]}")

# load contacts from file

def loadContacts():

if exists("contacts.txt"): # if file exists it will add all info to the lists

try:

with open("contacts.txt", "r") as file:

lines = file.readline()

for i in lines:

firstName, lastName, phoneNumber, address, postalCode = i.strip().spit(",")

firstNlist.append(firstName)

lastNlist.append(lastName)

phoneList.append(phoneNumber)

addressList.append(address)

postalCodeList.append(postalCode)

except:

print("error")

else:

print("no contact file saved we'll start fresh!") # will start with empty file if it can't find the file

# list all contacts

def listAll():

if len(firstNlist) == 0:

print("There are no contacts.")

else:

for i in range(len(firstNlist)):

print(f"{firstNlist[i]}, {lastNlist[i]}, {phoneList[i]}, {addressList[i]}, {postalCodeList[i]}")

# list contacts with last name first letter

def listbylastName(alpha):

for i in range(len(lastNlist)):

if lastNlist[i][0].lower() == alpha.lower():

print(f"{firstNlist[i]}, {lastNlist[i]}, {phoneList[i]}, {addressList[i]}, {postalCodeList[i]}")

else:

print("contact not found")

[7:33] **Main Loop**:

while True:

data = input("Welcome .. for a list of commands type (h)help or (q)uit: ")

if data.lower() == 'q':

saveContacts()

print("saving contacts...")

break

elif data.lower() == 'h':

print("NC - create new contact")

print("LA - lists all contacts")

print("FC - find contact")

print("DC - deletes contact")

print("LB - list by start letter of last name")

print("UP - update contact")

elif data.lower() == 'nc':

newContact()

elif data.lower() == 'la':

listAll()

elif data.lower() == 'fc':

searchContact = input("please enter a piece of information from the contact you are trying to find: ")

findContact()

elif data.lower() == 'dc':

searchContact = input("please enter first and last name of the contact you wish to delete: ")

deleteContact(searchContact)

elif data.lower() == 'lb':

alpha = input("please enter one letter of the last name you want to sort by: ")

listbylastName(alpha)

elif data.lower() == 'up':

firstName = ''

lastName = ''

updateContact(firstName, lastName)

else:

print("error: invalid command.")

[10:00] **Crash and bug test:**

I kept getting an error when running the program with a preexisted file, and when I ran the debugger on it, I realized I made a typo in the loadcontacts function where I wrote “spit” instead of “split”. I had an issue where it wasn’t writing one contact per line so I added a (\n) to the end of the savecontacts function and it seems to write one contact per line now. I had an error while testing out the update function where it would just end the function when I entered first and last name but it was because I had it indented incorrectly so after it broke out of the loop it wouldn’t go to the if statement. My deletecontact function wasn’t working and I kept getting an index error, so I changed it to be more flexible with a white loop and using two inputs.

def deleteContact(firstName, lastName):

found = False #assume false unless find in contacts

i = 0 # Initialize an index variable

while i < len(firstNlist):

if firstNlist[i].lower() == firstName.lower() and lastNlist[i].lower() == lastName.lower():

del firstNlist[i]

del lastNlist[i]

del phoneList[i]

del addressList[i]

del postalCodeList[i]

found = True

print("Contact successfully deleted.")

else:

i += 1

if not found:

print("name not found in contacts...")

[11:00] **Conclusion:**

Finally I will add some colors so it's not just black and white and boring. I had a lot of fun in the creation of this project, as it improved my skills in coding while using separate files to read and write over it. I feel like I had a lot of obstacles I had to deal with and I feel like I planned out my code a lot more this time around and it really benefited me in the implementation of the code.