--Pseudocode for "inventory.cpp"--

void displayMenu();

Display the menu and 4 choices

Prompt the user to enter the choice

bool isEmpty(file)

If the file is empty

Return true

Else

Return false

void newPart(list)

Prompt the user to enter the part's information

Input the information of the part into a new part object

Insert the new part object into the list

void findPart(list)

Prompt the user to enter a part's ID

Input the part's ID

Set list counter to the start of list

While the part's ID has not been found and list counter has not reached the end of the list

If the part's ID is equal to the ID the list counter points to

The part is found

Else

Increment the list counter

If the part is found

Print out the part's information

Else

Print out error message that the part is not found

void archiveBinary(list)

Create a binary file to output to named "binarySave.dat"

If the file cannot be created

Print error message

Set list counter to the start of the list

While list counter has not reached the end of the list

Copy the info of the part the list counter is pointing to into the file

Increment the list counter

void quitProgram(list)

Open the text file "textSave.txt" for outputting

If the file cannot be opened

Print error message

Set list counter to the start of the list

While list counter has not reached the end of the list

Copy the info of the part the list counter is pointing to into the file

Increment the list counter

Close the text file

int main()

Create a list to store part objects

Open the text file "textSave.txt" for inputting

If the file cannot be opened

Print error message

Exit the program

Else if the file is empty

Print error message

Else

Input the part's ID

While the end of the file has not been reached

Create a new Part object

Input the description of the part from the file

Input the price of the part from the file

Update the Part object's info using the info inputted from the file

Insert the Part object to the list

Read the next part's ID from the file

Close the text file

Call to function displayMenu()

Input user's choice

While user's choice is not equal to 'Q'

Case choice of

N: call to function newPart(list)

F: call to function findPart(list)

A: call to function archiveBinary(list)

default: print error message

End case

Call to function displayMenu()

Input user's choice

If user's choice is equal to 'Q'

Call to function quitProgram(list)

--End of the program--